

UNIVERSIDADE FEDERAL DO PARANÁ

IARA ZANDONAI DO NASCIMENTO

ABORDAGENS DETERMINÍSTICA E ESTOCÁSTICA PARA O PROBLEMA DE  
ROTEIRIZAÇÃO DE VEÍCULOS NA ENTREGA DE REFEIÇÕES

CURITIBA

2011

IARA ZANDONAI DO NASCIMENTO

ABORDAGENS DETERMINÍSTICA E ESTOCÁSTICA PARA O PROBLEMA DE  
ROTEIRIZAÇÃO DE VEÍCULOS NA ENTREGA DE REFEIÇÕES

Dissertação apresentada ao Curso de Pós-Graduação em Métodos Numéricos em Engenharia, Área de Concentração em Programação Matemática do Departamento de Matemática, Setor de Ciências Exatas e do Departamento de Construção Civil, Setor de Tecnologia, Universidade Federal do Paraná, como parte das exigências para a obtenção do título de Mestre em Ciências.

Orientadora: Prof.<sup>a</sup> Dr.<sup>a</sup> Neida Maria Patias Volpi

CURITIBA

2011

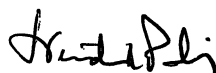
## TERMO DE APROVAÇÃO

**Iara Zandonai do Nascimento**

### **“Abordagens determinística e estocástica para o problema de roteirização de veículos na entrega de refeições”**

Dissertação aprovada como requisito parcial para obtenção do grau de Mestre no Curso de Pós-Graduação em Métodos Numéricos em Engenharia – Área de Concentração em Programação Matemática, Setores de Tecnologia e de Ciências Exatas da Universidade Federal do Paraná, pela seguinte banca examinadora:

Orientadora:



**Profª. Neida Maria Patias Volpi**, D.Eng. (orientadora)  
Programa de Pós-Graduação em Métodos Numéricos em Engenharia  
PPGMNE da UFPR



**Prof. Nelio Domingues Pizzolato**, D.Eng.  
PPGEP / PUC-RIO



**Profª. Izabel Cristina Zattar**, D.Eng.  
PPGEP / UFPR



**Profª. Luzia Vidal de Souza**, D.Sc.  
PPGMNE / UFPR

Curitiba, 05 de abril de 2011.

Dedico,

A Deus, primeiramente.

À minha mãe Leda, meu pai José e minha irmã Naira.

Aos meus amigos Fábio, Luciana, Marina e Sandro.

## **AGRADECIMENTOS**

Este espaço é dedicado àqueles que deram a sua contribuição para que esta dissertação fosse realizada. A todos eles deixo aqui o meu agradecimento sincero.

Agradeço a Deus, não apenas por mais essa conquista, mas por tudo que já me concedeu.

Aos meus pais, José e Leda, por todo amor, esforço e incentivo a mim dedicados.

À minha irmã Naira, pelo companheirismo em todos os momentos.

À minha orientadora Prof.<sup>a</sup> Neida pelos ensinamentos, paciência e dedicação sem os quais não seria possível concluir esse trabalho.

À Prof.<sup>a</sup> Izabel pela parceria e disponibilidade em ajudar.

A ASSEMCO, especialmente ao Gilberto, pelo apoio e parceria na realização deste trabalho.

Aos meus amigos Fábio, Luciana, Marina e Sandro por toda ajuda e incentivo nos momentos de dúvida, cansaço e desânimo.

À Maristela, secretária do PPGMNE, pela boa-vontade e carisma constantes.

Enfim, a todos que, direta ou indiretamente, contribuíram para a realização deste trabalho.

O caminho real da descoberta não está na procura de passagens  
novas, mas em possuir novos olhos.

Marcel Proust

## RESUMO

O problema de roteamento de veículos (PRV) é um dos mais estudados nas áreas de logística e otimização combinatória e consiste em determinar rotas de custo mínimo para uma frota de veículos, atendendo à demanda de um conjunto de clientes. Devido à complexidade computacional dos problemas de roteamento, torna-se inviável resolvê-los de forma exata, e em virtude disso, na maioria das vezes, a solução é obtida através de heurísticas. Mesmo sendo amplamente estudado o PRV carece de implementações práticas e nem sempre as soluções disponíveis resolvem satisfatoriamente os problemas das empresas, daí a necessidade de buscar soluções customizadas para cada problema. Dessa forma, o objetivo desse trabalho é desenvolver uma ferramenta que auxilie no processo de elaboração de roteiros de entrega de refeições feitas pela ASSEMCO - Associação dos Servidores Municipais de Colombo, localizada na região metropolitana de Curitiba, Paraná. A configuração atual dos roteiros de entrega não contempla todas as particularidades do problema, pois os clientes atendidos pela ASSEMCO possuem demanda variável e alguns destes precisam ser atendidos em horário fixo. Para resolução desse problema este trabalho apresenta duas abordagens: um programa computacional para o planejamento de rotas de entrega e um modelo estocástico que descreve o problema. Na abordagem determinística foi desenvolvido um programa em Visual Basic 6.0 que gera os roteiros de entrega através de métodos heurísticos de agrupamento e roteirização. Na modelagem estocástica do problema foram considerados a probabilidade de presença dos clientes, a probabilidade do veículo fazer o trajeto e o tempo de viagem entre os clientes, baseado na probabilidade de presença destes. Este modelo foi resolvido através do *software* Lingo 9.0 e através da meta-heurística *Ant Colony System*. Em relação à abordagem determinística, observou-se que os roteiros obtidos com a utilização do programa mostraram-se melhores em comparação com os roteiros normalmente praticados pela empresa. Em relação à abordagem estocástica, observou-se que o modelo proposto se adequou ao problema. Além disso, a meta-heurística *Ant Colony System* implementada para a resolução do modelo mostrou-se eficiente, visto que apresentou soluções de boa qualidade em tempo computacional viável.

Palavras-chave: Roteamento de veículos; Modelagem estocástica; Heurísticas; Otimização por colônia de formigas.

## **ABSTRACT**

The vehicle routing problem (VRP) is one of the most studied in logistics and combinatorial optimization and it consists in determining minimum cost routes for a fleet of vehicles, in order to serve the demand of a set of customers. Due to the computational complexity of routing problems, it becomes impossible to solve them optimally, so in most cases, the solution is achieved by heuristics implementation. Even widely studied, PRV lacks practical implementation and not always available solutions solve satisfactorily companies problems, hence the necessity of seeking customized solutions for each problem. Thus, the purpose of this research is to assist in the meal delivery routes process made by ASSEMCO - Colombo Municipal Servers Association, great Curitiba, in Paraná. The current configuration of delivery routes does not include all the details of the problem because the customers served by ASSEMCO have variable demand and some of them need to be attendend at fixed times. To solve this problem this research presents two approaches: a software for planning delivery routes and a stochastic model that describes the problem. In a deterministic approach, it was developed a program in Visual Basic 6.0 that generates the delivery route through heuristic methods for clustering and routing. In the stochastic modeling, based on customers presence probability, vehicle probability traversing the path, and the travel time between customers were considered. This model was solved by Lingo 9.0 software and through the meta-heuristic Ant Colony System. Concerning deterministic approach, it was observed that obtained routes by the program proved to be better compared to the route normally adopted by the company. Concerning stochastic approach, it was observed that the proposed model is suitable to the problem. In addition, the meta-heuristic Ant Colony System implemented for solving the model proved to be useful, because it produced good quality solutions in reasonable computational time.

Key-words: Vehicle routing; Stochastic modeling; Heuristics; Ant Colony Optimization.



## LISTA DE FIGURAS

FIGURA 2.1 – ESQUEMA DE SOLUÇÃO DE UM PRV .....	25
FIGURA 2.2 – EXPERIMENTO DAS DUAS PONTES (A)CAMINHOS DE MESMO COMPRIMENTO. (B) CAMINHOS DE COMPRIMENTOS DIFEREN- TES. ....	39
FIGURA 2.3 – RESULTADOS DO EXPERIMENTO DAS DUAS PONTES. (A) RESULTADOS PARA O CASO EM QUE OS CAMINHOS TÊM O MESMO COMPRIMENTO. (B) RESULTADOS PARA O CASO DE CAMINHOS DE COMPRIMENTOS DIFERENTES. ....	39
FIGURA 3.1 – MAPA DO MUNICÍPIO DE COLOMBO COM A LOCALIZAÇÃO DOS PONTOS DE DEMANDA E DEPÓSITO .....	50
FIGURA 3.2 – PERCENTUAL DE CLIENTES EM RELAÇÃO A REALIZAÇÃO DE PEDIDOS SEMANAIS .....	51
FIGURA 3.3 – DEMANDA MÉDIA DIÁRIA POR LOCAL .....	51
FIGURA 3.4 – DEMANDA DIÁRIA DO LOCAL DE ATENDIMENTO 39 .....	52
FIGURA 3.5 – DISTRIBUIÇÃO DE FREQUÊNCIA DE PEDIDOS DO LOCAL DE ATENDIMENTO 39 .....	52
FIGURA 4.1 – LOCALIZAÇÃO DAS SEMENTES PARA UMA DADA SIMULA- ÇÃO .....	56
FIGURA 4.2 – DETERMINAÇÃO DOS GRUPOS DE ATENDIMENTO .....	57
FIGURA 4.3 – INTERVALOS DE TEMPOS CONSIDERADOS NA HEURÍSTICA DO VIZINHO MAIS PRÓXIMO .....	58

FIGURA 4.4 – ROTEIRO A SER PERCORRIDO PELO VEÍCULO EM UMA DADA SIMULAÇÃO .....	60
FIGURA 4.5 – FLUXOGRAMA DE FUNCIONAMENTO DO PROGRAMA GE- RAROTA .....	61
FIGURA 4.6 – COMPARAÇÃO ENTRE AS ROTAS UTILIZADAS PELA ASSEMCO E ENCONTRADAS PELO PROGRAMA GERAROTA .....	62
FIGURA 5.1 – EXEMPLO DE CONFIGURAÇÃO DE ARCOS .....	67
FIGURA 5.2 – SOLUÇÃO EXATA DO MODELO PARA 10 NÓS - CENÁRIO 1 .	74
FIGURA 5.3 – DISTÂNCIA PERCORRIDA: SOLUÇÃO ESTOCÁSTICA E SO- LUÇÃO DETERMINÍSTICA .....	86

## LISTA DE TABELAS

TABELA 3.1	– DISTÂNCIA MÉDIA PERCORRIDA PELOS VEÍCULOS .....	53
TABELA 4.1	– DISTÂNCIA PERCORRIDA PELOS VEÍCULOS NO TRAJETO USADO PELA ASSEMCO E NA SIMULAÇÃO 1 .....	62
TABELA 4.2	– DISTÂNCIA PERCORRIDA PELOS VEÍCULOS NO TRAJETO USADO PELA ASSEMCO E NA SIMULAÇÃO 2 .....	62
TABELA 5.1	– SOLUÇÃO EXATA DO MODELO PARA 10 NÓS - CENÁRIO 1 .	73
TABELA 5.2	– SOLUÇÃO EXATA DO MODELO PARA 10 NÓS - CENÁRIO 2 .	73
TABELA 5.3	– SOLUÇÃO EXATA DO MODELO PARA 10 NÓS - CENÁRIO 3 .	74
TABELA 5.4	– SOLUÇÃO EXATA DO MODELO PARA 10 NÓS - CENÁRIO 4 .	74
TABELA 5.5	– RESUMO DAS SOLUÇÕES ENCONTRADAS PARA $N = 10$ ...	75
TABELA 5.6	– SOLUÇÃO PARA VALORES DE PENALIDADE ENTRE $[0, 10]$ .	75
TABELA 5.7	– SOLUÇÃO PARA VALORES DE PENALIDADE ENTRE $[0, 6]$ ..	76
TABELA 5.8	– SOLUÇÃO EXATA DO MODELO PARA 15 NÓS - CENÁRIO 1 .	76
TABELA 5.9	– SOLUÇÃO EXATA DO MODELO PARA 15 NÓS - CENÁRIO 2 .	76
TABELA 5.10	– SOLUÇÃO EXATA DO MODELO PARA 15 NÓS - CENÁRIO 3 .	77
TABELA 5.11	– SOLUÇÃO EXATA DO MODELO PARA 15 NÓS - CENÁRIO 4 .	77
TABELA 5.12	– RESUMO DAS SOLUÇÕES ENCONTRADAS PARA $N = 15$ ...	77
TABELA 5.13	– SOLUÇÃO EXATA DO MODELO PARA 20 NÓS - CENÁRIO 4 .	78
TABELA 5.14	– SOLUÇÃO ACS DO MODELO PARA 10 NÓS - CENÁRIO 1 ...	79

TABELA 5.15 – SOLUÇÃO ACS DO MODELO PARA 15 NÓS - CENÁRIO 2 ...	79
TABELA 5.16 – SOLUÇÃO ACS DO MODELO PARA 20 NÓS - CENÁRIO 4 ...	79
TABELA 5.17 – SOLUÇÃO ACS DO MODELO PARA 40 NÓS - CENÁRIO 1 - ROTA 1 .....	80
TABELA 5.18 – SOLUÇÃO ACS DO MODELO PARA 40 NÓS - CENÁRIO 1 - ROTA 2 .....	80
TABELA 5.19 – RESUMO DAS SOLUÇÕES OBTIDAS COM A META-HEURÍSTICA ACS .....	81
TABELA 5.20 – SOLUÇÕES EXATA E ACS PARA 10 NÓS - CENÁRIO 3 .....	82
TABELA 5.21 – SOLUÇÕES EXATA E ACS PARA 15 NÓS - CENÁRIO 4 .....	82
TABELA 5.22 – SOLUÇÕES EXATA E ACS PARA 20 NÓS - CENÁRIO 4 .....	83
TABELA 5.23 – COMPARAÇÃO ENTRE AS SOLUÇÕES EXATA E ACS .....	84
TABELA 5.24 – COMPARAÇÃO ENTRE OS TEMPOS DE EXECUÇÃO PARA AS SOLUÇÕES EXATA E ACS .....	84
TABELA 5.25 – COMPARAÇÃO ENTRE OS ROTEIROS ESTOCÁSTICO E DE- TERMINÍSTICO - 10 NÓS .....	85
TABELA 5.26 – COMPARAÇÃO ENTRE OS RESULTADOS ESTOCÁSTICOS E DETERMINÍSTICOS .....	86

## **LISTA DE QUADROS**

QUADRO 2.1	- TRABALHOS REFERENTES À SOLUÇÃO EXATA DO PCV ..	22
QUADRO 2.2	- TRABALHOS REFERENTES AO PRVE .....	31
QUADRO 2.3	- OUTROS TRABALHOS DE SOLUÇÃO DO PRVE .....	32

## SUMÁRIO

<b>1</b>	<b>INTRODUÇÃO</b>	<b>15</b>
1.1	OBJETIVOS	17
1.1.1	Objetivo geral	17
1.1.2	Objetivos específicos	17
1.2	JUSTIFICATIVA	17
1.3	ESTRUTURA DO TRABALHO	18
<b>2</b>	<b>REVISÃO DE LITERATURA</b>	<b>19</b>
2.1	PROBLEMA DO CAIXEIRO VIAJANTE	20
2.1.1	Métodos para resolução do problema do caixeiro viajante	21
2.1.1.1	Métodos exatos para o PCV	21
2.1.1.2	Métodos heurísticos para o PCV	22
2.1.1.2.1	Vizinho mais próximo	23
2.1.1.2.2	Melhoria 2-opt	23
2.2	PROBLEMA DE ROTEAMENTO DE VEÍCULOS	24
2.2.1	Problema de roteamento de veículos determinístico	26
2.2.1.1	Problema de roteamento de veículo capacitado	26
2.2.1.2	Problema de roteamento de veículo com janela de tempo	28
2.2.2	Problema de roteamento de veículos estocástico	31
2.3	PROBLEMA DE LOCALIZAÇÃO DE FACILIDADES	33

2.3.1 Algoritmo das p-medianas de Teitz e Bart .....	35
2.4 PROBLEMAS DE DESIGNAÇÃO .....	36
2.4.1 Algoritmo de Gillet e Johnson .....	37
2.5 OTIMIZAÇÃO POR COLÔNIA DE FORMIGAS .....	37
2.5.1 <i>Ant System</i> .....	40
2.5.2 <i>Ant Colony System</i> .....	42
2.5.3 <i>Multiple Ant Colony System</i> .....	44
2.6 TRABALHOS CORRELATOS .....	47
<b>3 DESCRIÇÃO DO PROBLEMA DE ENTREGA DE REFEIÇÕES .....</b>	<b>49</b>
<b>4 RESOLUÇÃO DO PROBLEMA DE ENTREGA DE REFEIÇÕES - CASO DE- TERMINÍSTICO .....</b>	<b>54</b>
4.1 CLUSTERIZAÇÃO .....	55
4.1.1 Determinação das sementes .....	55
4.1.2 Determinação dos grupos de atendimento .....	56
4.2 ROTEIRIZAÇÃO .....	57
4.2.1 Determinação do roteiro de entregas .....	58
4.2.2 Melhoria do roteiro encontrado .....	59
4.3 RESULTADOS .....	61
<b>5 RESOLUÇÃO DO PROBLEMA DE ENTREGA DE REFEIÇÕES - CASO ES- TOCÁSTICO .....</b>	<b>64</b>
5.1 IMPLEMENTAÇÃO DO ANT COLONY SYSTEM PARA RESOLUÇÃO DO MO- DELO PRVER .....	70
5.2 RESULTADOS .....	71

5.2.1 Método exato aplicado ao modelo PRVER .....	72
5.2.2 Meta-heurística <i>Ant Colony System</i> aplicada ao modelo PRVER .....	78
5.3 COMPARAÇÃO DOS RESULTADOS EXATOS E HEURÍSTICOS .....	81
5.4 ANÁLISE COMPARATIVA ENTRE OS CASOS DETERMINÍSTICO E ESTOCÁSTICO .....	85
<b>6 CONSIDERAÇÕES FINAIS E SUGESTÕES PARA TRABALHOS FUTUROS</b>	<b>87</b>
<b>REFERÊNCIAS</b> .....	<b>89</b>
<b>APÊNDICE A – JANELA PRINCIPAL - PROGRAMA GERAROTA</b> .....	<b>94</b>
<b>APÊNDICE B – JANELA DE LOCAIS CADASTRADOS - PROGRAMA GERAROTA</b> .....	<b>95</b>
<b>APÊNDICE C – JANELA COM O TRAJETO SUGERIDO - PROGRAMA GERAROTA</b> .....	<b>96</b>
<b>ANEXO A – DISTRIBUIÇÕES DE PROBABILIDADE</b> .....	<b>97</b>



## 1 INTRODUÇÃO

A eficiência nos processos logísticos de distribuição de mercadorias e serviços é um fator de grande influência na competitividade de uma empresa. Dessa forma, a melhoria destes serviços tem ganhado atenção especial nas últimas décadas. Tanto as instituições privadas quanto as públicas visam alcançar um bom nível de satisfação do cliente a um baixo custo operacional. Os custos de transporte têm peso significativo na formação desse custo operacional e, segundo Ballou (2001), podem chegar a atingir dois terços do custo total da operação, por isso problemas na área de distribuição, tais como os problemas de roteamento, têm sido amplamente estudados.

O problema de roteamento de veículos (PRV) foi introduzido por Dantzig e Ramser, em 1959, e tornou-se um dos mais estudados nas áreas de logística e otimização combinatória em virtude de sua aplicação prática. O PRV consiste basicamente em entregar produtos, para um dado conjunto de clientes, usando uma frota de veículos com capacidade limitada. Devido a sua dinamicidade o PRV apresenta várias extensões, dentre elas está o problema de roteamento de veículos com janela de tempo (PRVJT). Esse problema tem como objetivo determinar as rotas de custo mínimo para uma frota de veículos de mesma capacidade, atendendo à demanda de um conjunto de clientes, para os quais o atendimento somente é possível dentro de intervalo de tempo determinado, chamado janela de tempo.

No PRV padrão os elementos que constituem o sistema são determinísticos, ou seja, as demandas, os tempos, os custos e os locais são conhecidos antecipadamente. Entretanto, na maioria dos problemas reais, os elementos do sistema são bastante variáveis e por isso o problema deve ser considerado estocástico.

Em termos computacionais os problemas de roteirização de veículos são classificados como *NP-hard*, ou seja, possuem ordem de complexidade exponencial. O que

significa dizer que o esforço computacional para a sua resolução cresce exponencialmente com o tamanho do problema. Diante disso, os métodos exatos de solução tornam-se inviáveis, pois exigem elevado tempo computacional. Nesse sentido, os métodos heurísticos surgem como alternativa para a resolução desse tipo de problema, pois oferecem boas soluções em tempo computacional menor.

Mesmo sendo amplamente estudado, o problema de roteamento de veículos varia em função de restrições e parâmetros e nem sempre os *softwares* disponíveis no mercado resolvem satisfatoriamente os problemas das empresas, daí a necessidade de buscar soluções customizadas para cada problema.

Sendo assim, a finalidade desse trabalho é desenvolver uma ferramenta eficiente que venha auxiliar no processo de elaboração de roteiros de entrega de refeições feitas pela ASSEMCO - Associação dos Servidores Municipais de Colombo, localizada na região metropolitana de Curitiba, Paraná.

A ASSEMCO é uma entidade associativa que presta diversos serviços aos funcionários públicos do município de Colombo, dentre esses serviços, destaca-se o de entrega de refeições. Alguns pedidos de entrega são agendados, mas em sua maioria os pedidos são feitos diariamente e em virtude disso a demanda torna-se bastante variável, bem como os locais a serem atendidos. Além disso, o intervalo de tempo em que os clientes devem ser atendidos é pequeno e determinados lugares preferem receber as refeições em horários fixos. As rotas de entrega são elaboradas manualmente baseadas na experiência do funcionário encarregado do processo. Entretanto essa prática nem sempre é a mais eficaz porque nem todos os locais precisam ser atendidos diariamente. Isso muitas vezes acarreta perda da qualidade da refeição, a qual permanece muito tempo dentro do veículo, e insatisfação dos clientes que não recebem o produto no horário desejado.

## 1.1 OBJETIVOS

### 1.1.1 Objetivo geral

O objetivo geral desse trabalho é estudar e aplicar heurísticas e meta-heurísticas a um problema de roteirização de veículos com janela de tempo e demanda variáveis.

### 1.1.2 Objetivos específicos

Esse trabalho tem como objetivos específicos:

- Implementar a meta-heurística *Ant Colony Optimization*(ACO) para resolver o modelo proposto e comparar os resultados com a solução exata do problema, quando possível.
- Desenvolver um *software* de roteirização que sirva como ferramenta para o planejamento de rotas de entrega feitas pela ASSEMCO.
- Apresentar um modelo matemático para o problema de roteamento de veículos com janela de tempo com elementos estocásticos incorporados na modelagem.

## 1.2 JUSTIFICATIVA

Embora sendo amplamente estudados, os problemas de roteamento ainda carecem de implementação prática. Existem soluções disponíveis no mercado, entretanto estas nem sempre atendem às particularidades de cada empresa. Segundo Hall e Partyka (1997), na área de roteirização de veículos uma solução para um determinado tipo de problema e dados pode não ser adequada para outro problema similar. Sendo assim é necessário encontrar soluções específicas para cada situação.

Segundo Gendreau (2010), em relação aos problemas de roteamento de veículos com informações variadas ou não-determinísticos, existem poucos trabalhos considerando tempos de serviço e tempos de viagem estocásticos, sendo essas informações incertas na maioria dos problemas de roteamento. Ainda segundo Gendreau

(2010), modelos e técnicas de solução dos problemas de roteamento estocástico podem ser úteis para tratar problemas que não são realmente estocásticos mas exibem estrutura semelhante.

### 1.3 ESTRUTURA DO TRABALHO

Este trabalho está organizado em mais cinco capítulos.

No capítulo 2 é feita uma revisão de literatura a respeito do problema de roteamento de veículos e suas variações, e das principais técnicas de solução.

O capítulo 3 descreve o problema de entrega de refeições e como este é tratado atualmente.

No capítulo 4 é apresentado o programa implementado para a solução do problema de entrega de refeições e os resultados obtidos com sua utilização.

O capítulo 5 apresenta um modelo estocástico para o problema em questão, uma implementação da meta-heurística *Ant Colony Sytem* para a resolução do modelo proposto e as soluções encontradas.

O capítulo 6 conclui essa dissertação e apresenta recomendações para trabalhos futuros.

## 2 REVISÃO DE LITERATURA

Neste capítulo são abordados alguns conceitos, formulações matemáticas e técnicas de solução de alguns problemas de roteamento. Além disso são citadas as principais referências e trabalhos correlatos.

A seção 2.1 apresenta o problema do caixeiro viajante, o modelo precursor dos modelos de roteirização. Na seção 2.2 é descrito o problema de roteamento de veículos e algumas de suas variações. Nas seções 2.3, 2.4 e 2.5 são apresentadas as técnicas de solução do problema de roteamento de veículos utilizadas neste trabalho, sendo elas, algoritmo das  $p$ -medianas de Teitz e Bart, algoritmo de designação de Gillet e Johnson e otimização por colônia de formigas.

Antes, porém, serão apresentadas alguns noções básicas e notações sobre grafos importantes para esse trabalho.

Segundo Goldberg e Luna (2005), um grafo é uma estrutura formada por um conjunto de nós (vértices) e um conjunto de arcos (arestas) conectando todos ou alguns desses nós. Sendo  $N$  o conjunto de nós, e  $A$  o conjunto dos arcos entre os nós, um grafo pode ser denotado por  $G = (N, A)$ . Se forem associados valores numéricos aos nós ou arcos de um grafo  $G$ , tem-se um grafo ponderado.

Define-se por subgrafo de  $G$  um grafo cujo conjunto dos nós é um subconjunto do conjunto de nós de  $G$  e cujo conjunto de arcos é também um subconjunto do conjunto de arcos de  $G$ .

Quando o sentido das ligações entre os nós é importante, ou seja, quando os arcos possuem uma direção, o grafo  $G$  é denominado orientado. Um grafo é dito completo quando existir ao menos uma ligação associada a cada par de nós, e é dito simétrico se para cada arco  $(u, v)$  existe um correspondente arco reverso  $(v, u)$ .

Ainda segundo Goldbarg e Luna (2005), um caminho em um grafo é uma sequência de nós em que cada dois nós consecutivos são ligados por um arco. Um caminho é simples se não tem nós repetidos. Em um grafo ponderado o comprimento de um caminho é a soma dos pesos associados aos seus arcos. Um circuito é um caminho que começa e termina no mesmo nó. Um circuito em que os nós não se repetem, exceto o final e o inicial que são os mesmos, é denominado circuito elementar. Um circuito hamiltoniano é um circuito elementar que passa por todos os nós do grafo  $G$ .

## 2.1 PROBLEMA DO CAIXEIRO VIAJANTE

O Problema do Caixeiro Viajante(PCV) é um dos mais estudados na área de Otimização Combinatória. Sua popularidade, segundo Goldbarg e Luna (2005), deve-se principalmente a sua larga aplicabilidade, sua relação com outros modelos e também com a grande dificuldade em encontrar soluções exatas.

Dado um grafo  $G = (N, A)$  o PCV consiste em encontrar um circuito hamiltoniano de custo mínimo. No grafo  $G = (N, A)$ ,  $N = \{1, \dots, n\}$  representa o conjunto de nós,  $A = \{(i, j) : i, j \in N, i \neq j\}$  representa o conjunto de arcos associados às conexões entre os nós, e  $c_{ij}$  representa o custo associado ao arco  $(i, j)$ .

Existem várias formulações para esse problema. Nas equações (2.1)-(2.5) encontra-se a formulação proposta por Dantzig, Fulkerson e Johnson (1954):

### MODELO-PCV:

$$\text{minimizar} \quad \sum_{i=1}^n \sum_{j=1}^n c_{ij} x_{ij} \quad (2.1)$$

sujeito a

$$\sum_{i=1}^n x_{ij} = 1 \quad \forall j \in N \quad (2.2)$$

$$\sum_{j=1}^n x_{ij} = 1 \quad \forall i \in N \quad (2.3)$$

$$\sum_{i,j \in S}^n x_{ij} \leq |S| - 1 \quad \forall S \subset N \quad (2.4)$$

$$x_{ij} \in \{0, 1\} \quad \forall i, j \in N \quad (2.5)$$

As restrições (2.2) e (2.3) garantem que cada nó seja visitado uma única vez. O conjunto  $S$  é um subgrafo de  $G$ , e  $|S|$  indica o número de nós desse subgrafo. Dessa forma as restrições (2.4) garantem a não-existência de sub-rotas. O tipo de variável é indicado nas restrições (2.5). A variável binária  $x_{ij}$  assume valor 1 caso o arco  $(i, j)$  esteja na rota percorrida pelo caixeiro, e assume valor 0 caso contrário.

### 2.1.1 Métodos para resolução do problema do caixeiro viajante

Nesta seção estão descritos alguns métodos para a resolução do PCV.

#### 2.1.1.1 Métodos exatos para o PCV

Uma maneira intuitiva de se encontrar soluções ótimas para o PCV seria testar todas as possibilidades de caminhos, entretanto sabe-se que essa é uma estratégia inviável em virtude do número de combinações possíveis.

Segundo Bodin *et al.* (1983) soluções exatas para o PCV podem ser encontradas utilizando-se o algoritmo de *Branch-and-Bound* ou Programação Inteira Mista.

O quadro 2.1 apresentado em Goldberg e Luna (2005), mostra alguns trabalhos referentes à solução exata do Problema do Caixeiro Viajante.

Ano	Pesquisador	Trabalho
1954	Dantzig et al.	Trabalho referência para o PCV
1972	Christofides e Elion	Métodos exatos
1973	Laporte e Norbert	Métodos exatos
1980	Carpaneto e Toth	Critérios para algoritmos <i>Branch-and-Bound</i>
1980	Crowder e Padberg	<i>Branch-and-Bound</i>
1981	Balas e Christofides	Restrições lagrangeanas para o PCV
1985	Fleischmann	Algoritmo com uso de plano de cortes
1991	Miller e Pekny	Métodos exatos
1992	Gendreau et al.	Procedimentos de pós-otimização
1995	Jünger et al.	Relaxações e <i>Branch-and-Cut</i>
1998	Applegate et al.	<i>Branch-and-Cut</i>
2000	Jünger et al.	Cortes
2001	Applegate et al.	Cortes
2002	Focacci et al.	Relaxação em restrições globais
2004	Marcotte et al.	Programação multinível

QUADRO 2.1 - Trabalhos referentes à solução exata do PCV  
 FONTE: Goldberg e Luna (2005)

Uma biblioteca criada por Reinelt (1990) contém várias instâncias para o PCV que vêm sendo testadas na literatura. Ainda existem instâncias não solucionadas nesta biblioteca, entretanto é notório o avanço que se tem alcançado na solução do PCV nos últimos anos: Dantzig, Fulkerson e Johnson (1954) resolveram o problema para 49 cidades, Applegate *et al.* (2004) resolveram o problema para 24978 cidades.

No entanto, apesar do avanço tecnológico, principalmente em relação ao poder computacional disponível, mesmo os melhores algoritmos exatos tem dificuldade em encontrar as soluções ótimas para problemas maiores ou em um tempo computacional razoável. Sendo assim, uma alternativa para resolução desse tipo de problema são os métodos heurísticos podem oferecer soluções aproximadas de boa qualidade.

#### 2.1.1.2 Métodos heurísticos para o PCV

Devido a complexidade computacional dos problemas de Otimização Combinatória, métodos exatos de solução são inviáveis para grandes instâncias. Nestes casos métodos heurísticos são indicados, pois geram soluções próximas à solução ótima em tempo computacional adequado.

Existem inúmeros métodos heurísticos para a resolução do Problema do Caixeiro Viajante. A seguir estão descritos os que foram implementados neste trabalho.



#### 2.1.1.2.1 Vizinho mais próximo

O algoritmo do vizinho mais próximo é uma heurística de construção de rotas que consiste em escolher como próximo movimento o nó mais próximo ainda não visitado. O desenvolvimento dessa heurística é normalmente atribuído a Cover e Hart (1967).

O funcionamento dessa heurística está descrito no Algoritmo 1:

**PASSO 1:** Marque todos os nós como não visitados.

**PASSO 2:** Escolha algum nó  $s$  como ponto de partida, e marque  $s$  como visitado. Faça  $v = s$ .

**PASSO 3:** Repita enquanto houver nós a serem visitados:

Escolha o nó  $w$  não-visitado mais próximo de  $v$ .

Marque  $w$  como visitado.

Vá de  $v$  para  $w$  e faça  $v = w$ .

**PASSO 4:** Vá do nó  $v$  para o nó  $s$ .

#### **Algoritmo 1:** Vizinho mais próximo

O algoritmo do vizinho mais próximo é uma heurística gulosa, ou seja, sempre escolhe a melhor opção entre as disponíveis. Heurísticas construtivas gulosas, apesar de serem rápidas, tendem a gerar soluções de baixa qualidade. Para melhorar a qualidade da solução encontrada por essas heurísticas outros procedimentos devem ser utilizados.

#### 2.1.1.2.2 Melhoria 2-opt

Os procedimentos de melhoria de rotas visam melhorar uma solução já existente, visto que as heurísticas construtivas não costumam gerar soluções ótimas.

Lin (1965) introduziu os procedimentos de melhoria 2-opt e 3-opt. A heurística de melhoramento 2-opt consiste na troca de um par de nós que compõe a solução por

outros dois nós. Após essa troca a nova solução é avaliada. Se o custo for menor que o custo da solução anterior, a nova solução é tomada como solução ótima atual para o problema. Esse processo continua até que não haja mais trocas a serem realizadas.

A heurística 2-opt está descrita no Algoritmo 2:

**PASSO 1:** Encontre uma solução inicial através de uma heurística construtiva.

**PASSO 2:** Remova dois arcos do roteiro atual e reconecte os nós que formam esses dois arcos, modificando as ligações.

**PASSO 3:** Se o resultado da troca for melhor, substitua o roteiro atual pelo novo roteiro encontrado. Senão, mantenha o roteiro atual e escolha dois outros arcos. Volte ao Passo 2 até que não hajam mais trocas a serem efetuadas.

#### **Algoritmo 2:** Heurística 2-opt

No procedimento 3-opt, três arcos são removidos do roteiro. Assim como na 2-opt, os caminhos resultantes são reconectados de forma que gerem uma solução diferente.

## 2.2 PROBLEMA DE ROTEAMENTO DE VEÍCULOS

Problema de roteamento de veículos (PRV) é o nome dado a uma classe de problemas que consiste em atender certo número de clientes dispersos geograficamente utilizando uma frota de veículos que parte de um depósito. O objetivo do problema é encontrar um conjunto de rotas de custo mínimo satisfazendo as seguintes restrições básicas:

- Cada rota inicia e termina no depósito.
- Cada cliente deve ser visitado apenas uma vez.

A figura 2.1 ilustra um esquema de solução para um PRV. Este exemplo compreende três rotas, ou seja, três veículos que atendem 20 clientes.

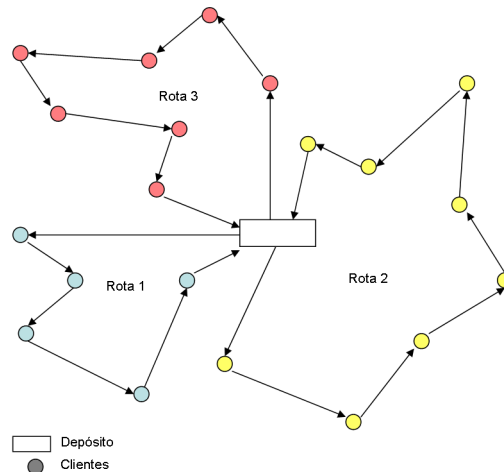


FIGURA 2.1: Esquema de solução de um PRV

FONTE: O autor(2011)

Dentre as aplicações práticas do PRV, Goldbarg e Luna (2005) destacam:

- Distribuição de produtos (DANTZIG; RAMSER, 1959);
- Distribuição de jornais((GOLDEN; MAGNANTI; NGUYEN, 1977);
- Transporte escolar (NEWTON; THOMAS, 1974);
- Recolhimento de lixo (BELTRAMI; BODIN, 1974);
- Roteamento de helicópteros (TIMLIN; PULLEYBLANK, 1990);
- Distribuição de alimentos (BARTHOLDI *et al.*, 1983);
- Roteamento de satélites (LEE *et al.*, 2003).

A partir do problema clássico de roteirização é possível definir um conjunto de problemas de roteamento, através da alteração ou inclusão de restrições e parâmetros. Algumas dessas variações são encontradas na literatura. Como exemplo de algumas tem-se:

- Tempo fixo para atender os clientes ou realizar o processo de entregas;
- Frota homogênea ou heterogênea;

- Um ou múltiplos depósitos;
- Natureza dos parâmetros utilizados: determinísticos, estocásticos ou dinâmicos;
- Operação a ser realizada: coleta, entrega ou ambas.

Essas variações levam a diferentes classificações do PRV. As mais relevantes a esse trabalho estão descritas nas seções 2.2.1 e 2.2.2.

### 2.2.1 Problema de roteamento de veículos determinístico

O problema de roteamento de veículos determinístico assume que todos os elementos do sistema são conhecidos antecipadamente, ou seja, os clientes, as demandas, os custos, entre outros são conhecidos *a priori*. As rotas obtidas para o problema não são alteradas, independentemente de qualquer evento que possa vir a ocorrer.

#### 2.2.1.1 Problema de roteamento de veículo capacitado

A abordagem mais simples de um PRV é conhecida como Problema de Roteamento de Veículos Capacitado (PRVC) e considera, além das restrições básicas (cada rota inicia e termina no depósito e cada cliente é atendido apenas uma vez), a capacidade de carga de cada veículo.

Uma formulação para esse problema foi proposta por Fisher e Jaikumar (1981) e está descrita nas equações (2.6) a (2.14).

Seja um grafo não direcionado  $G = (N, A)$ , onde  $N = V \cup \{0\}$ ,  $V = \{1, \dots, n\}$  é o conjunto de nós que representam os clientes e 0 é o nó que representa o depósito, e  $A = \{(i, j) : i, j \in N, i \neq j\}$  é o conjunto de arcos associados às conexões entre os nós. Cada cliente  $i$  possui uma demanda  $d_i$  e a cada arco  $(i, j) \in A$  está associado um custo de utilização  $c_{ij}$ . Um conjunto  $K$  de veículos idênticos está situado no depósito, onde  $K = \{1, \dots, m\}$ , e cada veículo  $k \in K$  tem capacidade  $Q$ .

As variáveis de decisão utilizadas no modelo são:

$$x_{ij}^k = \begin{cases} 1 & \text{se o arco } (i, j) \text{ está na rota percorrida pelo veículo } k, \forall k \in K \text{ e } \forall (i, j) \in A \\ 0 & \text{caso contrário} \end{cases}$$

$$y_i^k = \begin{cases} 1 & \text{se o cliente } i \text{ é atendido pelo veículo } k, \forall k \in K \text{ e } \forall i \in V \\ 0 & \text{caso contrário} \end{cases}$$

### MODELO-PRVC:

$$\text{minimizar} \quad \sum_{k \in K} \sum_{(i,j) \in A} c_{ij} x_{ij}^k \quad (2.6)$$

sujeito a

$$\sum_{k \in K} y_i^k = 1 \quad i \in V \quad (2.7)$$

$$\sum_{k \in K} y_i^k = m \quad i = 0 \quad (2.8)$$

$$\sum_{i \in V} d_i y_i^k \leq Q \quad k \in K \quad (2.9)$$

$$\sum_{i \in N} x_{ij}^k = y_j^k \quad j \in V \text{ e } k \in K \quad (2.10)$$

$$\sum_{j \in N} x_{ij}^k = y_i^k \quad i \in V \text{ e } k \in K \quad (2.11)$$

$$\sum_{(i,j) \in S} x_{ij}^k \leq |S| - 1 \quad S \subseteq \{2, \dots, n\} \text{ e } k \in K \quad (2.12)$$

$$x_{ij}^k \in \{0, 1\} \quad i, j \in N \text{ e } k \in K \quad (2.13)$$

$$y_i^k \in \{0, 1\} \quad i, j \in V \text{ e } k \in K \quad (2.14)$$

A função objetivo (2.6) minimiza o custo total de utilização das rotas. As restrições (2.7) asseguram que cada cliente seja atendido por um único veículo.

As restrições (2.8) garantem que todos os  $m$  veículos passem pelo depósito ( $i=0$ ). Neste modelo o número de veículos é um parâmetro fixo. No caso em que o número de veículos é variável, retiram-se as restrições (2.8) e atribui-se um custo  $c_v$  para cada veículo usado. Se o custo  $c_v$  de se utilizar cada veículo for grande, o modelo priorizará a redução do número de veículos sobre o custo das rotas.

As restrições (2.9) impõem que a demanda total de cada rota não exceda a capacidade do veículo. As restrições (2.10) e (2.11) são relativas ao fluxo em redes, e garantem que os veículos não interrompam suas rotas em um cliente. Já as restrições (2.12) asseguram a não-existência de sub-rotas.

As restrições (2.13) e (2.14) indicam o tipo de variável.

Abordagens exatas para resolução do PRVC podem ser encontradas em: Crowder e Padberg (1980) que utilizaram a técnica *branch-and-bound*, Laporte, Nobert e Desrochers (1985) que apresentaram o primeiro método exato com uma abordagem baseada em planos de cortes, Christofides, Mingozzi e Toth (1981a) que basearam-se em relaxação lagrangeana e dualidade, e ainda Padberg e Rinaldi (1991) e Ralphs *et al.* (2003) que obtiveram bons resultados utilizando a técnica de *branch-and-cut*.

A literatura sobre o PRVC mostra diferentes abordagens para encontrar soluções aproximadas para o problema. Dentre outras, pode-se citar Colônia de formigas ((BULLNHEIMER; HARTL; STRAUSS, 1999), Busca Tabu (BARBAROSOGLU; OZGUR, 1999) e (REIMANN; STUMMER; DOERNER, 2002)), Algoritmo evolucionário (PRINS, 2004) e Algoritmo Genético (BAKER; AYECHEW, 2003) .

#### 2.2.1.2 Problema de roteamento de veículo com janela de tempo

O problema de roteamento de veículos com janela de tempo (PRVJT) é uma extensão do PRVC, na qual o atendimento a cada cliente somente é possível dentro de intervalo de tempo pré-determinado, chamado janela de tempo. Janelas de tempo ditas *soft* podem ser violadas, enquanto que no caso de janelas de tempo rígidas (*hard*) só permitem que o cliente seja atendido durante sua janela. Se o veículo chega antes do início da janela, ele deve esperar, e a chegada ao cliente depois do fim de sua janela é estritamente proibida.

Para formular o PRVJT considera-se que a cada arco  $(i, j) \in A$  está associado um tempo de viagem  $t_{ij}$ , e a cada cliente  $i \in V$  está associada uma janela de tempo  $[a_i, b_i]$  e um tempo de atendimento  $r_i$ . Além disso considere a variável adicional:

$s_i^k$  = instante em que o veículo  $k$  começa a servir o cliente  $i$ ,  $\forall k \in K$  e  $\forall i \in V$ , e as seguintes restrições adicionais:

- Se o veículo  $k$  viaja do nó  $i$  para o nó  $j$ , então não pode chegar em  $j$  antes de  $s_i^k + r_i + t_{ij}$ . Esta restrição pode ser representada pela equação não linear dada em (2.15):

$$x_{ij}^k(s_i^k + r_i + t_{ij} - s_j^k) \leq 0, \quad (i, j) \in A \quad \text{e} \quad k \in K \quad (2.15)$$

Sendo  $M$  uma constante arbitrária, a restrição (2.15) na forma linearizada é equivalente a:

$$s_i^k + r_i + t_{ij} - M(1 - x_{ij}^k) \leq s_j^k, \quad (i, j) \in A \quad \text{e} \quad k \in K \quad (2.16)$$

- Todas as janelas de tempo devem ser respeitadas:

$$a_i \leq s_i^k \leq b_i, \quad i \in V \quad \text{e} \quad k \in K \quad (2.17)$$

- O tempo de viagem de uma rota não pode exceder o limite de tempo total  $T$  percorrido pelo veículo:

$$\sum_{(i,j) \in A} t_{ij} x_{ij}^k \leq T, \quad k \in K \quad (2.18)$$

- As variáveis  $s_i^k$  devem ser maiores ou iguais a zero:

$$s_i^k \geq 0, \quad i \in V \quad \text{e} \quad k \in K \quad (2.19)$$

O modelo matemático para o PRVJT consiste da função objetivo dada pela equação (2.6) sujeito às restrições (2.7)-(2.14) e (2.16)-(2.19).

Abordagens exatas e heurísticas têm sido amplamente estudadas para obter soluções para o PRVJT. As primeiras técnicas de solução podem ser encontradas em Kolen, Kan e Trienekens (1987), Desrochers (1988) e Solomon e Desrochers (1988). Kolen, Kan e Trienekens (1987) introduziram o primeiro método para encontrar a solução exata para o PRVJT, através da extensão do algoritmo de caminho mínimo de Christofides, Mingozzain e Toth (1981b).

Os principais métodos exatos utilizados na solução do PRVJT se baseiam em técnicas de relaxação, planos de corte e geração de colunas. Desrochers, Desrosiers e Solomon (1992) usaram o método de geração de colunas, Kohl e Madsen (1997) desenvolveram algoritmo baseado em relaxação Lagrangeana e Cook e Rich (1999) utilizaram algoritmos baseados em planos de cortes. Ainda em abordagens exatas existem os trabalhos recentes de Feillet (2004) e Rousseau, Gendreau e Pesant (2004).

Devido à complexidade computacional do PRVJT muitos autores têm se dedicado ao estudo de heurísticas e metaheurísticas como técnicas de solução, já que essas, ao contrário dos métodos exatos, exploram apenas uma parte do espaço de soluções, mas fornecem soluções de boa qualidade a um baixo custo computacional.

Solomon (1987) adaptou a heurística de varredura desenvolvida por Gillet e Miller (1974) para o problema de roteamento com janela de tempo. Potvin e Rousseau (1993) introduziram uma versão paralela à heurística proposta por Solomon (1987). Ioannou (2001) baseou-se também em Solomon (1987) e criou novos critérios de seleção e inserção de nós. Bramel e Simchi-Levi (1993) introduziram um algoritmo de duas fases para a solução do PRVJT. Muitas heurísticas têm sido aplicadas como uma pós-otimização a fim de melhorar a solução encontrada. Exemplos dessas aplicações no PRVJT podem ser encontrados em Russel (1995) e Bräsy (2003).

Abordagens metaheurísticas também têm sido aplicadas ao problema de roteamento com janela de tempo como Busca Tabu (BRÄSSY; GENDREAU, 2002), Simulated Annealing ((CHIANG; RUSSEL, 1996),(CZECH; CZARNAS, 2002) e (LI; LIM, 2003)), Colônia de formigas (GAMBARDELLA; TAILLARD; AGAZZI, 1999) , Algoritmos evolucionários ((HOMBERGER; GEHRING, 1999), (BERGER; BARKAOUI; BRÄSSY, 2001) e (TING; HUANG, 2004))e GRASP -Greedy Randomized Adaptive Search Procedure (KONTORAVDIS; BARD, 1995).



### 2.2.2 Problema de roteamento de veículos estocástico

O problema de roteamento de veículos estocástico (PRVE) é uma variação do PRV na qual um ou mais elementos são gerados de forma aleatória ou probabilística.

Alguns exemplos comuns de PRVE são:

- Problema de roteamento de veículos com demanda estocástica: segundo Gendreau, Laporte e Séguin (1996) é um dos mais estudados de todos os PRVEs. Nessa variação as demandas dos clientes são variáveis randômicas, geralmente assumidas como independentes e seguindo alguma distribuição de probabilidade conhecida.
- Problema de roteamento de veículos com clientes estocásticos: o conjunto de clientes a ser visitado não é conhecido com certeza, cada cliente possui uma probabilidade  $p_i$  de estar presente, ou seja, cada cliente possui uma probabilidade  $p_i$  de possuir demanda.
- Problema de roteamento de veículos com tempo de viagem estocástico: neste caso o tempo necessário para um veículo deslocar-se do cliente  $i$  para o cliente  $j$  é gerado de forma probabilística. Esse tipo de problema pode estar relacionado a diversos fatores, tais como o clima, trecho em obras, acidentes, entre outros.

Segundo Larsen (2000), as abordagens de resolução para problemas de roteamento estocásticos podem ser classificadas como: otimização em tempo real e *a priori*.

As abordagens de otimização em tempo real constroem as rotas durante sua execução e baseiam-se na suposição implícita de que as informações são reveladas ao longo do tempo em que os veículos realizam suas rotas.

Nos métodos de resolução *a priori*, uma solução inicial é obtida a partir das probabilidades de existência dos clientes, demanda, tempos de viagem, entre outros. Dentro dessa classe destaca-se a chamada modelagem em dois estágios ou modelagem

com recurso. Um recurso é uma ação necessária quando a solução inicial torna-se infactível no momento em que todos os parâmetros são conhecidos.

No caso do problema de roteamento de veículos com demanda estocástica, a demanda de cada cliente está associada a uma probabilidade. Quando as demandas se tornam conhecidas o veículo pode seguir a rota definida *a priori* ignorando os pontos onde não houve demanda ou “pular” tais pontos. Quando uma rota falha, ou seja, a capacidade do veículo é excedida, o veículo retorna ao depósito e inicia novamente sua rota no próximo cliente. Esse recurso de voltar ao depósito gera um custo que é adicionado no segundo estágio. O objetivo do problema então, é minimizar o custo da solução no primeiro estágio e o custo total do recurso utilizado no segundo estágio. Segundo Lorenzi (2003), uma boa solução *a priori* deve determinar rotas de baixo custo que sejam pouco influenciadas pela existência ou não de demanda.

O quadro 2.2 apresentado em Goldberg e Luna (2005), mostra alguns trabalhos referentes ao problema de roteamento de veículos estocástico:

Ano	Pesquisador	Trabalho
1992	Cock e Rinnooy Kan	Revisão do problema
1993	Bertsimas e Van Ryzin	Roteamento no plano euclidiano com múltiplos veículos
1993	Belgacem et al.	Roteamento estocástico com split delivery
1995	Bertsimas et al.	Vários algoritmos
1995	Gendreau et al.	Algoritmos para o caso da demanda e cliente estocásticos
1996a	Gendreau et al.	Revisão de literatura
1996b	Gendreau et al.	Algoritmo tabu
1996	Cheung e Power	Modelos e algoritmos
1996	Teodorovic e Pavkovic	Aplicação da teoria dos conjuntos nebulosos
1997	Ong et al.	Demanda estocástica e janela de tempo
2000	Yang et al.	Roteamento estocástico com estoque
2001	Secomandi	Abordagem rollout
2001	Taniguchi et al.	Roteamento com tempo de percurso estocástico
2003	Bianchi et al.	Meta-heurísticas
2003	Verweij et al.	Método estatístico de amostragem
2003	Kenyon e Morton	<i>Branch-and-cut</i> e amostragem Monte Carlo

QUADRO 2.2 - Trabalhos referentes ao PRVE

FONTE: Goldberg e Luna (2005)

O quadro 2.3 apresenta outros trabalhos de pesquisa relacionados ao PRVE:

Pesquisador e ano	Metodologia
Mark e Guo (2004)	Algoritmo genético
Tan, Cheong e Goh (2007)	Computação evolucionária
Haugland, Ho e Laporte (2007)	Busca tabu e heurísticas multi-start
Pavone <i>et al.</i> (2009)	Modelo para roteamento dinâmico e estocástico com janela de tempo
Secomandi e Margot (2009)	Processo de Markov
Bard (2010)	Simulação de Monte Carlo
Rei, Gendreau e Soriano (2010)	Monte Carlo e Local Branching

QUADRO 2.3 - Outros trabalhos de solução do PRVE  
FONTE: O autor(2011)

## 2.3 PROBLEMA DE LOCALIZAÇÃO DE FACILIDADES

Problemas de localização de facilidades têm como objetivo determinar a melhor localização de facilidades, considerando-se os clientes que podem ser servidos. Esses problemas podem ser divididos em:

- Problemas de localização de centros, cujo objetivo é escolher os locais de forma a minimizar a distância até o ponto mais crítico a ser atendido, ou seja, minimizar a maior distância a ser percorrida por qualquer cliente aos centros escolhidos.
- Problemas de localização de medianas, cujo objetivo é escolher locais de modo a minimizar a soma de todas as distâncias de cada ponto de demanda à mediana mais próxima.

O problema das  $p$ -medianas consiste em determinar a localização de  $p$  facilidades (medianas) em um conjunto (grafo) de  $n$  facilidades possíveis,  $n > p$ , de forma a minimizar a soma total das distâncias percorridas de cada ponto de demanda até a mediana mais próxima.

Seja um grafo  $G = (N, A)$  simétrico, onde  $N$  representa o conjunto de nós e  $A$  representa o conjunto de ligações entre os nós. Seja  $D = (d_{ij})$  a matriz de distâncias entre todos os nós de  $G$  e  $X = (x_{ij})$  a matriz de alocação, onde:

$$x_{ij} = \begin{cases} 1 & \text{se o nó } j \text{ é alocado ao nó } i \\ 0 & \text{caso contrário} \end{cases}$$

Além disso,

$$x_{ii} = \begin{cases} 1 & \text{se o nó } i \text{ é uma mediana} \\ 0 & \text{caso contrário} \end{cases}$$

Sendo  $p$  o número de facilidades (medianas) a serem localizadas, o problema das  $p$ -medianas pode ser descrito pelas equações (2.20) a (2.24):

**MODELO-P-MEDIANAS:**

$$\text{minimizar} \quad \sum_{i=1}^n \sum_{j=1}^n d_{ij} x_{ij} \quad (2.20)$$

sujeito a

$$\sum_{i=1}^n x_{ij} = 1 \quad j = 1, \dots, n \quad (2.21)$$

$$\sum_{i=1}^n x_{ii} = p \quad (2.22)$$

$$x_{ij} \leq x_{ii} \quad i, j = 1, \dots, n \quad (2.23)$$

$$x_{ij} \in \{0, 1\} \quad i, j = 1, \dots, n \quad (2.24)$$

A função objetivo é definida na equação (2.20). As restrições em (2.21) asseguram que cada nó seja designado a apenas uma mediana. A restrição (2.22) garante que existem exatamente  $p$  medianas. As restrições em (2.23) garantem que as alocações só podem ser feitas nas medianas. As restrições em (2.24) correspondem ao tipo de variável.

Esse modelo é dito não-capacitado, pois um número qualquer de pontos de demanda podem ser alocados em uma mediana. No caso de problemas em que os pontos candidatos a mediana tem uma capacidade limitada deve-se incluir ainda a restrição:

$$\sum_{j=1}^n q_j x_{ij} \leq Q_i x_{ii}, \quad i = 1, \dots, n \quad (2.25)$$

onde  $q_j$  é a demanda do ponto  $j$  e  $Q_i$  é capacidade da mediana  $i$ .

O problema das  $p$ -medianas pertence à classe dos problemas *NP-hard* e por isso sua solução exata torna-se inviável em problemas de grande porte. Em 2.3.1 está descrito um método aproximado de solução para o problema que foi implementado nesse trabalho.

### 2.3.1 Algoritmo das $p$ -medianas de Teitz e Bart

O algoritmo das  $p$ -medianas proposto por Teitz e Bart (1968), é um método aproximado para a determinação de  $p$  medianas e baseia-se na substituição de nós.

O objetivo do problema das  $p$ -medianas é encontrar um conjunto  $V_p \in N$  para o qual o número de transmissão  $\sigma(V_p)$  seja mínimo. O número de transmissão de um conjunto  $S$  é:

$$\sigma(S) = \sum_{i=1}^n \left\{ w_i \min_{j \in S} [d(i, j)] \right\} \quad (2.26)$$

Onde  $w_i$  é o peso associado ao nó  $i$ ,  $\min[d(i, S)]$  é a menor distância do nó  $i$  aos nós do conjunto  $S$ , sendo  $S$  o conjunto candidato a solução  $V_p$  do problema das medianas.

No algoritmo 3 estão descritos os passos executados pelo método proposto por Teitz e Bart:

**PASSO 1:** Selecione um conjunto  $S$ , com  $|S| = p$  para formar uma solução inicial para o problema das  $p$ -medianas.

**PASSO 2:** Rotule todos os nós  $i \notin \{N - S\}$  como “não analisados”.

**PASSO 3:** Enquanto existirem nós “não analisados” em  $N - S$ , faça:

Selecione um nó  $i \in N - S$ , “não analisado”, e calcule a variação  $\Delta_{ij}$  do número de transmissão, para todo  $j \in S$ :

$$\Delta_{ij} = \sigma(S) - \sigma(S \cup \{i\} - \{j\})$$

Faça  $\Delta_{ij\_max} = \max \{\Delta_{ij}\}$ , para todo  $\Delta_{ij}$  calculado.

Se  $\Delta_{ij\_max} > 0$  faça  $S \leftarrow (S \cup \{i\} - \{j\})$  e insira  $j$  em  $\{V - S\}$ . Rotule  $j$  como “analisado” e inicie o Passo 3 novamente.

Caso contrário rotule  $i$  como “analisado” e inicie o Passo 3 novamente.

Não existindo mais nós a serem analisados, apresente o conjunto  $S$  obtido.

**PASSO 4:** Utilize o conjunto  $S$  obtido no Passo 3 e inicie novamente pelo Passo 2. Se, durante a execução do Passo 3, houver alguma modificação no conjunto  $S$ , volte ao Passo 2. Caso contrário, PARE e apresente o conjunto  $S$  como uma solução aproximada para o problema das  $p$ -medianas.

### Algoritmo 3: Teitz e Bart

## 2.4 PROBLEMAS DE DESIGNAÇÃO

O problema de designação consiste em alocar cada um dos pontos de demanda a cada uma das facilidades (medianas), de maneira a agrupar os pontos mais próximos. O objetivo do problema é designar cada um dos pontos de demanda a cada uma das medianas de forma a minimizar a distância total.

### 2.4.1 Algoritmo de Gillet e Johnson

O algoritmo de designação proposto por Gillet e Johnson (1974) busca resolver problemas de múltiplas facilidades, relacionando cada ponto de demanda à facilidade mais próxima. Este processo está descrito no algoritmo 4:

**PASSO 1:** Calcule a distância entre cada nó  $i$  ainda não designado até cada uma das medianas (facilidades).

**PASSO 2:** Para cada nó  $i$  do passo anterior, defina  $L_1(i)$  e  $L_2(i)$ , respectivamente a 1ª e 2ª medianas mais próximas ao ponto  $i$ , com distâncias iguais a  $D_1(i)$  e  $D_2(i)$ .

**PASSO 3:** Calcule a razão:  $r(i) = \frac{D_1(i)}{D_2(i)}$

Ordene os nós  $i$  de acordo com os valores de  $r(i)$  em ordem crescente. Esta lista determina a ordem em que os nós serão designados a cada uma das medianas, obedecendo-se as capacidades de cada local.

**PASSO 4:** Percorra a lista do passo anterior, designando os nós  $i$  às medianas mais próximas, até que a capacidade de alguma delas fique esgotada. Neste caso, retire todos os nós  $i$  já designados e a mediana com capacidade esgotada e volte ao passo 1.

**Algoritmo 4:** Gillet e Johnson

## 2.5 OTIMIZAÇÃO POR COLÔNIA DE FORMIGAS

Os chamados algoritmos de formigas pertencem ao grupo de Algoritmos Evolutivos inspirados na biologia e são derivados da observação do comportamento de formigas reais. A ideia por trás desses algoritmos é a de que embora algumas espécies de formigas sejam praticamente cegas elas são capazes de locomover-se em ambientes complexos. Essa capacidade provém de um processo chamado estigmer-

gia (do inglês *stigmergy*), um mecanismo de coordenação indireta entre agentes. O princípio é que o traço deixado no meio ambiente por um agente influencia o desempenho de outros agentes. Esse traço, é na verdade, uma substância química chamada feromônio, que quando captada por animais de uma mesma espécie permitem um reconhecimento mútuo dos indivíduos. As formigas depositam no chão o feromônio e traçam um caminho do ninho até uma fonte de comida. Seguindo o traço de feromônio as formigas podem seguir o caminho para a comida descoberto pelas formigas precedentes.

Um experimento realizado por Deneubourg *et al.* (1990), conhecido como o experimento das duas pontes, mostrou a capacidade de otimização das formigas.

No primeiro experimento descrito na figura 2.2(a) o ninho das formigas é conectado a uma fonte de comida por uma ponte de dois caminhos de mesmo comprimento. No início dos testes não há feromônio em nenhum dos dois caminhos. Como as formigas não tem uma preferência elas escolhem com a mesma probabilidade algum dos caminhos, conforme resultado apresentado na figura 2.3(a). No segundo experimento descrito na figura 2.2(b) o comprimento de um dos caminhos é aumentado. Neste caso, após algum tempo as formigas escolhem usar somente o menor caminho, conforme resultado apresentado na figura 2.3(b). Isso ocorre devido à quantidade de feromônio depositada no caminho: grandes quantidades de feromônio estimulam mais formigas a escolherem o caminho novamente.

Esse comportamento das formigas de deixar um rastro a ser seguido por outras formigas é a motivação da metodologia de Otimização por Colônia de Formigas, do inglês *Ant Colony Optimization* (ACO).

Segundo Dorigo, Caro e Gambardella (1999), os algoritmos de otimização por colônia de formigas possuem as seguintes características:

- As formigas cooperam entre si para realizar uma tarefa. Soluções de boa qualidade só são possíveis com a interação de todos os agentes da colônia.



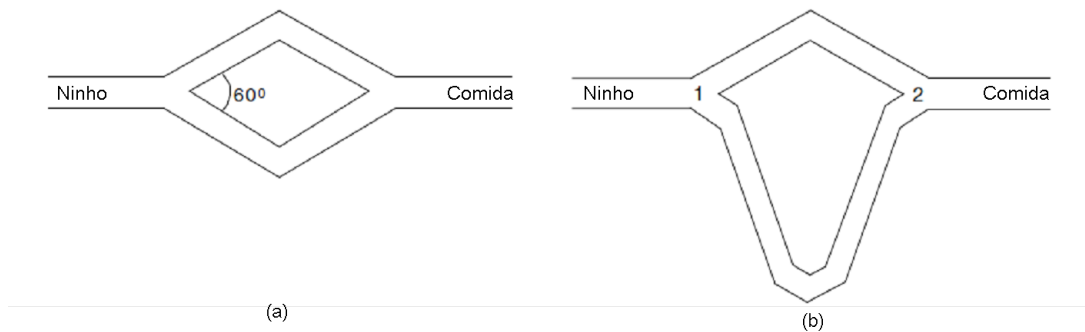


FIGURA 2.2: Experimento das duas pontes (a)Caminhos de mesmo comprimento. (b) Caminhos de comprimentos diferentes.

FONTE: Dorigo e Stützle (2004)

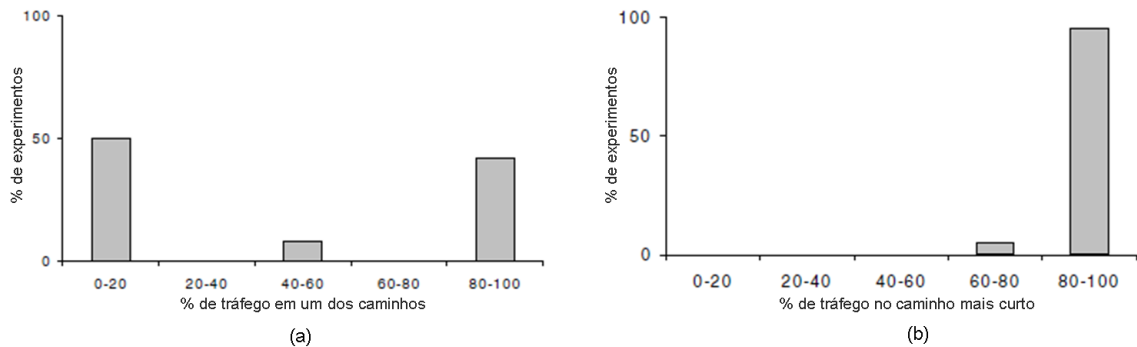


FIGURA 2.3: Resultados do experimento das duas pontes. (a) Resultados para o caso em que os caminhos têm o mesmo comprimento. (b) Resultados para o caso de caminhos de comprimentos diferentes.

FONTE: Dorigo e Stützle (2004)

- A comunicação entre as formigas é indireta, ocorre somente através do depósito e seguimento do rastro de feromônio. Com o passar do tempo ocorre a evaporação do feromônio o que causa o esquecimento do passado recente e permite que as formigas explorem novas soluções.
- As formigas artificiais possuem memória na qual armazenam informações sobre o caminho percorrido.
- Uma formiga pode mover-se de um ponto para qualquer outro em sua vizinhança factível. Esse movimento é escolhido através de uma regra de decisão probabilística.

### 2.5.1 Ant System

O *Ant System*(AS) foi o primeiro algoritmo ACO implementado. Foi proposto por Dorigo, Maniezzo e Coloni (1996) e sua primeira aplicação prática foi ao problema do caixeiro viajante.

O *Ant System* inicia (instante  $t = 0$ ) com  $m$  formigas distribuídas aleatoriamente pelas  $n$  cidades. A cada iteração as formigas escolhem a próxima cidade a ser visitada através de uma regra de decisão probabilística que leva em consideração a atratividade e a “visibilidade” do arco. A atratividade  $\tau_{ij}$  representa a quantidade de feromônio depositada no arco  $(i, j)$ , ou seja, indica quantas formigas escolheram esse mesmo arco. A visibilidade  $\eta_{ij}$  do arco indica que quanto menor o comprimento do arco mais desejável ele é, também denominada de valor heurístico.

Para gerar uma rota factível uma formiga deve visitar todas as  $n$  cidades, transições entre cidades já visitadas são proibidas até que uma rota seja completada. Essas proibições são armazenadas na memória da formiga, denominada lista tabu.

A probabilidade da formiga  $h$  mover-se da cidade  $i$  para a cidade  $j$ , no instante  $t$ , é dada por:

$$p_{ij}^h(t) = \begin{cases} \frac{[\tau_{ij}(t)]^\alpha \cdot [\eta_{ij}]^\beta}{\sum_{k \notin M_h} [\tau_{ik}(t)]^\alpha \cdot [\eta_{ik}]^\beta}, & \text{se } j \notin M_h \\ 0, & \text{caso contrário} \end{cases} \quad (2.27)$$

Onde:

$\tau_{ij}$  representa a atratividade do arco  $(i, j)$ .

$\eta_{ij} = \frac{1}{d_{ij}}$  é a visibilidade do arco  $(i, j)$ , e  $d_{ij}$  representa o comprimento do arco  $(i, j)$ .

$M_h$  é a lista tabu da formiga  $h$  e armazena as cidades que já foram visitadas.

$k$  é o nó factível a ser escolhido pela formiga.

$\alpha$  e  $\beta$  são parâmetros que controlam a influência de  $\tau_{ij}$  e  $\eta_{ij}$ , respectivamente.

$\alpha, \beta \in [0, 1]$ .

No algoritmo AS, uma iteração representa os  $m$  movimentos realizados pelas  $m$  formigas no intervalo  $(t, t + 1)$ , então a cada  $n$  iterações, o que é denominado um ciclo, cada formiga completa uma rota (*tour*).

No final de cada ciclo, ou seja, quando as soluções já foram construídas, acontece a atualização do feromônio. Cada formiga  $h$  deposita uma quantidade  $\Delta\tau_{ij}^h(t)$  em cada arco pertencente a sua rota:

$$\Delta\tau_{ij}^h(t) = \begin{cases} \frac{1}{L^h(t)} & \text{se } (i, j) \in T^h(t) \\ 0, & \text{caso contrário} \end{cases} \quad (2.28)$$

Onde:

$T^h(t)$  é a rota feita pela formiga  $h$  com comprimento  $L^h(t)$ .

Uma taxa de evaporação é associada ao feromônio, a fim de reduzir a quantidade de feromônio em cada arco. Isso é feito para evitar que as formigas fiquem presas em ótimos locais.

A atualização do feromônio é calculada através da expressão:

$$\tau_{ij}(t + n) = (1 - \rho)\tau_{ij}(t) + \Delta\tau_{ij}(t) \quad (2.29)$$

Onde:

$\rho$  é a taxa de evaporação,  $\rho \in (0, 1]$ .

$\Delta\tau_{ij}(t) = \sum_{h=1}^m \Delta\tau_{ij}^h(t)$  é a soma das contribuições individuais de cada formiga que passou pelo arco  $(i, j)$ .

O processo AS pode ser descrito pelo algoritmo 5:

Repita

    Posicione aleatoriamente  $m$  formigas em  $n$  nós.

    Para nó= 1 até  $n$  faça:

        Para  $formiga = 1$  até  $m$  faça:

            Escolha o próximo nó para se mover através da equação

(2.27).

        fim Para.

    fim Para.

    Calcule o tamanho da rota percorrida por cada formiga.

    Aplique o depósito de feromônio de cada formiga através da equação

(2.28).

    Faça a atualização do feromônio através da equação (2.29).

Até que a condição de parada seja satisfeita.

#### **Algoritmo 5: *Ant System***

#### 2.5.2 *Ant Colony System*

O *Ant Colony System* (ACS) foi proposto por Gambardella e Dorigo (1996) e resultou de um aperfeiçoamento feito do algoritmo AS. Uma diferença entre o ACS e o AS, diz respeito a forma de atualização do feromônio, que no algoritmo ACS ocorre tanto localmente quanto globalmente. Entretanto, a principal diferença é em relação a regra de decisão das formigas.

No ACS, a regra de decisão das formigas envolve um critério denominado pseudo-aleatório. Nesse critério, um parâmetro  $q_0$  é introduzido. Esse parâmetro define o poder de exploração das formigas. A regra de decisão é dada por:

$$j = \begin{cases} \arg \max_{j \notin M_h} \left\{ [\tau_{ij}(t)]^\alpha \cdot [\eta_{ij}(t)]^\beta \right\}, & \text{se } q \leq q_0 \\ \text{índice definido pela expressão (2.27),} & \text{caso contrário} \end{cases} \quad (2.30)$$

Onde:

$\tau_{i,j}(t)$  é a quantidade de feromônio associada ao arco  $(i, j)$ .

$\eta_{ij}(t)$  é o valor heurístico.

$\alpha$  é o parâmetro que controla a influência do traço de feromônio e  $\beta$  é o parâmetro que indica a importância do valor heurístico.

$q$  é um valor escolhido randomicamente com probabilidade uniforme em  $[0, 1]$  e  $q_0 (0 \leq q_0 \leq 1)$  é um parâmetro que determina se a formiga fará uma decisão randômica ou gulosa.

$M_h$  é a lista tabu da formiga  $h$  e armazena as cidades que já foram visitadas.

A regra de decisão apresentada em (2.30) compara um número aleatório  $q$  a um parâmetro  $q_0$ , sempre que a formiga necessita escolher o próximo nó para se mover. Se o valor de  $q$  for menor que  $q_0$ , o algoritmo funcionará de forma gulosa, ou seja de acordo com a menor distância. Caso contrário, a solução é gerada de forma probabilística, segundo a expressão (2.27).

Enquanto constroem uma rota, as formigas depositam feromônio nos arcos que visitam, sendo  $\rho$  a taxa de evaporação de feromônio, a atualização local é dada pela seguinte equação:

$$\tau_{ij}(t+1) = (1 - \rho)\tau_{ij}(t) + \rho\tau_0 \quad (2.31)$$

Onde  $\tau_0$  indica a concentração inicial de feromônio.

A atualização global do feromônio é feita de forma a privilegiar os arcos pertencentes à melhor solução global. No final de cada ciclo, somente a formiga que encontrou a melhor solução irá depositar feromônio. A atualização é calculada da seguinte maneira:

$$\tau_{ij}(t+n) = (1 - \rho)\tau_{ij}(t) + \frac{\rho}{L_{melhor}}, \quad \forall (i, j) \in T_{melhor} \quad (2.32)$$

Onde:

$L_{melhor}$  é o comprimento da melhor rota encontrada desde o início do algoritmo.

$T_{melhor}$  é a melhor rota encontrada desde o início do algoritmo.

### 2.5.3 *Multiple Ant Colony System*

Outros algoritmos foram desenvolvidos baseados no ACS, um exemplo deles é o MACS-VRPTW *Multiple Ant Colony System for Vehicle Routing Problem with Time Windows*. O MACS-VRPTW foi proposto por Gambardella, Taillard e Agazzi (1999) e é uma extensão do ACS aplicada ao problema de roteamento de veículos com janela de tempo.

Como o objetivo do PRVJT é minimizar tanto o número de veículos utilizados quanto a distância total percorrida por eles, o MACS-VRPTW consiste em usar duas colônias de formigas para que cada uma delas busque a minimização de um dos objetivos do problema. As duas colônias cooperam trocando informações através da atualização de feromônio.

A primeira colônia, chamada ACS-VEI, tenta minimizar o número de veículos, a outra colônia, denominada ACS-TIME, tenta minimizar, para um dado número de veículos, o tempo de viagem total. A idéia proposta por Gambardella, Taillard e Agazzi (1999) é a de que os algoritmos sejam executados em computação paralela. Entretanto, para facilitar a implementação do algoritmo, o MACS-VRPTW pode ser executado em ciclos, nos quais as colônias são ativadas de forma alternada.

De acordo com Gambardella, Taillard e Agazzi (1999), a colônia ACS-TIME possui o mesmo objetivo do tradicional ACS, que é calcular a menor rota possível. Já a colônia ACS-VEI procura por soluções factíveis maximizando o número de clientes visitados. Como ACS-VEI propõe uma solução com um veículo a menos do que a melhor solução encontrada até o momento, isso normalmente gera soluções infactíveis, ou seja, alguns clientes deixam de ser visitados. O objetivo da ACS-VEI é encontrar a

melhor solução (geralmente infactível) com o maior número de clientes visitados.

No algoritmo MACS-VRPTW o depósito e todas as suas conexões com os clientes, são duplicadas pelo número de vezes equivalente ao número de veículos disponíveis, visto que nos problemas de roteamento de veículos considera-se que todas as rotas iniciam e terminam no depósito. As distâncias entre as cópias do depósito são iguais a zero.

ACS-VEI e ACS-TIME utilizam um procedimento chamado *new\_active\_ant* para gerar as rotas das formigas. Nesse procedimento cada formiga inicia sua rota a partir da escolha randômica de uma cópia do depósito e, a cada passo, move-se para um nó ainda não visitado, que não viole as restrições de janela de tempo e capacidade dos veículos. O conjunto de nós disponíveis, no caso de uma formiga não estar em um depósito duplicado, também inclui os depósitos duplicados ainda não visitados. Uma formiga localizada no nó  $i$  escolhe probabilisticamente o próximo nó  $j$  a ser visitado através da equação (2.30).

A atratividade de cada arco é calculada considerando-se o tempo de viagem  $t_{ij}$  entre os nós  $i$  e  $j$ , a janela de tempo  $[a_j, b_j]$  associada ao nó  $j$ , e o número de vezes  $IN_j$  que o nó  $j$  não foi inserido na solução encontrada pela colônia ACS-VEI, sendo atribuído o valor zero para ACS-TIME.

Sendo *delivery\_time* o momento em que o nó  $j$  poderá ser atendido por um veículo, *tempo\_atual* o tempo de serviço do nó atual, *delta\_time* o intervalo de tempo que se passará do momento atual até o início do serviço no nó em questão,  $d_{ij}$  e  $t_{ij}$  a distância e o tempo de viagem entre os nós  $i$  e  $j$  respectivamente, a atratividade  $\eta_{ij}$  do arco  $(i, j)$  é dada por:

$$delivery\_time_j = \max(tempo\_atual_i + t_{ij}, a_j) \quad (2.33)$$

$$delta\_time_j = delivery\_time_j - tempo\_atual_i \quad (2.34)$$

$$d_{ij} = \text{delta\_time}_j * (b_j - \text{tempo\_atual}_i) \quad (2.35)$$

$$d_{ij} = \max(1, d_{ij} - IN_j) \quad (2.36)$$

$$\eta_{ij} = \frac{1}{d_{ij}} \quad (2.37)$$

Cada vez que uma formiga se move de um nó para outro, é feita a atualização local do feromônio de acordo com a expressão(2.31). No caso da ACS-VEI, se no fim da fase construtiva, a solução estiver incompleta, ou seja, sem a inclusão de alguns clientes, estes serão inseridos nas rotas por ordem decrescente de quantidade de demanda. Para cada cliente procura-se a melhor inserção factível.

Tanto na ACS-TIME quanto na ACS-VEI ao final de cada ciclo, ou seja, quando uma solução é encontrada, a atualização global do feromônio é feita através da equação (2.32).

Seja  $\psi_{gb}$  uma solução factível para o PRVJT, encontrada pela heurística do vizinho mais próximo.  $\psi_{gb}$  será melhorada pelo MACS-VRPTW. Este processo está descrito no algoritmo 6:



$\psi_{gb}$  solução inicial factível encontrada pela heurística do vizinho mais próximo.

Repita

$v = \text{número de veículos utilizados em } \psi_{gb}.$

Ative ACS-VEI( $v - 1$ ).

Ative ACS-TIME( $v$ ).

Enquanto ACS-VEI e ACS-TIME estão ativas

Espera por uma solução melhorada  $\psi$  de ACS-VEI e ACS-TIME.

$\psi_{gb} \leftarrow \psi$

Se número de veículo em  $\psi_{gb} < v$  então

Encerre ACS-TIME e ACS-VEI.

Fim enquanto.

Até que a condição de parada seja satisfeita.

#### **Algoritmo 6: MACS-VRPTW**

## 2.6 TRABALHOS CORRELATOS

Diversos trabalhos têm sido propostos abordando diferentes técnicas de solução para o problema de roteamento e suas variações. Os métodos exatos, embora encontrem soluções ótimas, são inviáveis para problemas de grande porte, as heurísticas surgem então como uma alternativa de se encontrar soluções boas em tempo computacional viável.

Embora existam várias soluções para problemas práticos de roteirização, segundo Bräysy *et al.* (2008) o problema de entrega de refeições é uma tarefa bastante realizada mas que ainda apresenta poucas técnicas de otimização implementadas para sua resolução. Ainda segundo Bräysy *et al.* (2008), os problemas de entrega de refeições, quando resolvidos de forma eficaz, apresentam uma redução de custos bastante significativa.

Para Johnson, Roehrig e Yildiz (2009), coordenar entregas de refeições é uma tarefa difícil devido às variações da demanda e às limitações de tempo para realizar as entregas a fim de garantir uma refeição de qualidade.

Algoritmos baseados em *savings*(economias) foram implementados por Atkinson (1990) e Suensilpong (2007). Atkinson (1990) estudou um problema de roteamento de veículos com janela de tempo para programação de veículos que entregam refeições em escolas. Suensilpong (2007) apresentou um estudo de caso no qual as entregas são realizadas por motocicletas, neste caso a capacidade do veículo é bastante limitada.

Bräysy *et al.* (2008) consideraram o problema de entrega de refeições como um problema de múltiplos caixeiros viajantes. No caso estudado a capacidade do veículo não era restritiva e foi utilizado um *software* comercial para resolução do problema.

Johnson, Roehrig e Yildiz (2009) apresentaram um estudo de caso no qual as refeições são entregues nas casas de pacientes idosos e enfermos. Para resolução do problema Johnson, Roehrig e Yildiz (2009) utilizaram Algoritmo Genético para agrupar as demandas e traçar as rotas de entrega.

Considerando que o problema de entrega de refeições ainda desperta interesse, no capítulo 3 é apresentado um problema dessa natureza. Os capítulos 4 e 5 apresentam como foram implementadas as heurísticas e meta-heurísticas descritas neste capítulo para a resolução deste problema.

### **3 DESCRIÇÃO DO PROBLEMA DE ENTREGA DE REFEIÇÕES**

A ASSEMCO - Associação dos Servidores Municipais de Colombo foi fundada em 1985 e é a maior entidade associativa de representação no município de Colombo, Paraná. A associação presta serviços de assistência social, médica, odontológica, farmacêutica e alimentícia a seus associados.

Em relação à assistência alimentícia, a ASSEMCO possibilita aos servidores utilizar o restaurante localizado no centro da cidade ou receber a refeição no local de trabalho. O serviço de entrega de refeições constitui um grande problema, pois apesar de que alguns pedidos de entrega serem previamente agendados, em sua grande maioria, os pedidos são feitos diariamente e os roteiros de entrega são elaborados manualmente, a partir da experiência do funcionário encarregado da realização do serviço. Essa prática mostrou-se inadequada em face ao crescente número de reclamações e da insatisfação dos usuários em relação a qualidade da refeição, muitas vezes entregue fria ou com embalagem danificada, devido ao fato desta permanecer por um longo período dentro do veículo.

O intervalo para a realização das entregas das refeições é de aproximadamente duas horas. Os pedidos são recebidos até às 10h30min e as entregas são realizadas no período das 11h30min às 13h30min. Entretanto, determinados lugares preferem receber as refeições em horários fixos, como escolas por exemplo. Os horários preferenciais para as refeições são entre 12h e 13h, entretanto, nas escolas esse horário é reduzido para o intervalo entre 12h e 12h30min.

A figura 3.1 apresenta a localização dos pontos de demanda e também do restaurante (depósito) onde são preparadas as refeições e de onde partem os veículos. Os locais de destinos são escolas, postos de saúde, creches entre outros estabelecimentos da Prefeitura Municipal de Colombo.

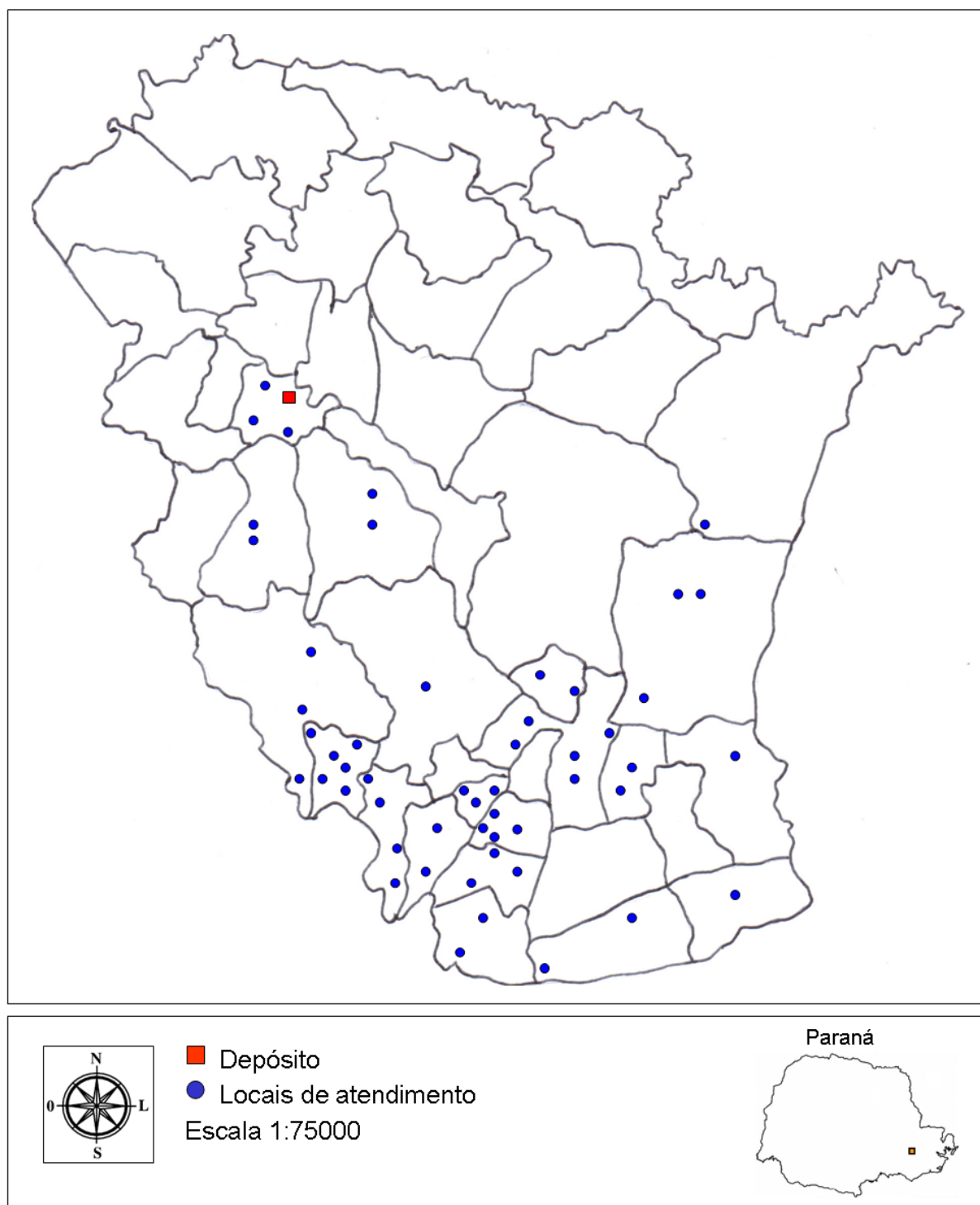


FIGURA 3.1: Mapa do município de Colombo com a localização dos pontos de demanda e depósito

FONTE: O autor(2011)

Como grande parte dos pedidos é feita diariamente a demanda torna-se bastante variável, bem como os locais de entrega a serem atendidos. Analisando-se

o percentual dos clientes em relação à realização de pedidos feitos semanalmente (FIGURA 3.2) tem-se que 41% dos clientes realiza pedidos pelo menos 4 vezes na semana, 31% utiliza o serviço de 2 a 3 vezes na semana e 22% dos clientes de 0 a 1 vez na semana. Os dados são referentes ao ano de 2009.

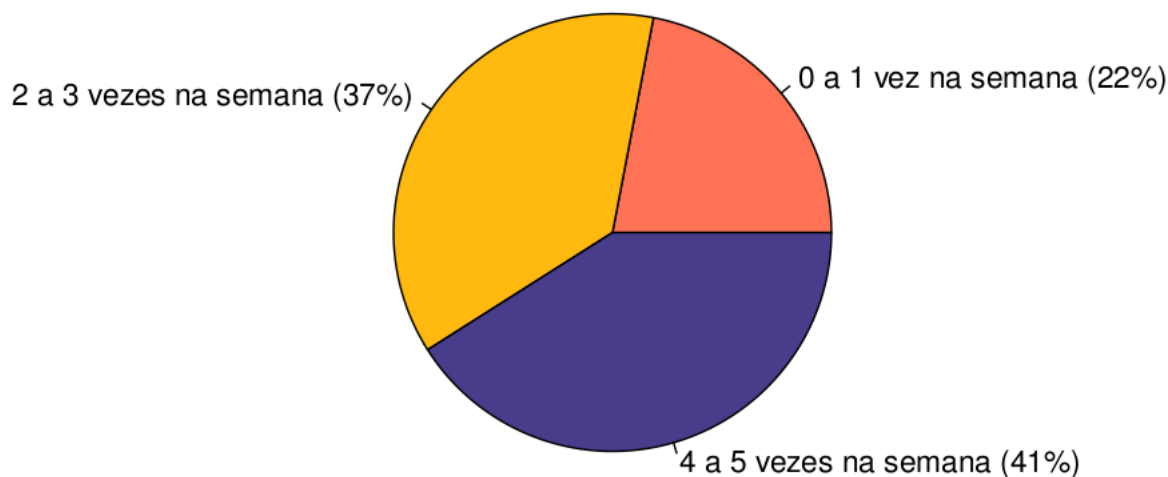


FIGURA 3.2: Percentual de clientes em relação a realização de pedidos semanais

FONTE: O autor(2011)

A média de pedidos diária no ano de 2009 para cada local de atendimento está apresentada na figura 3.3.

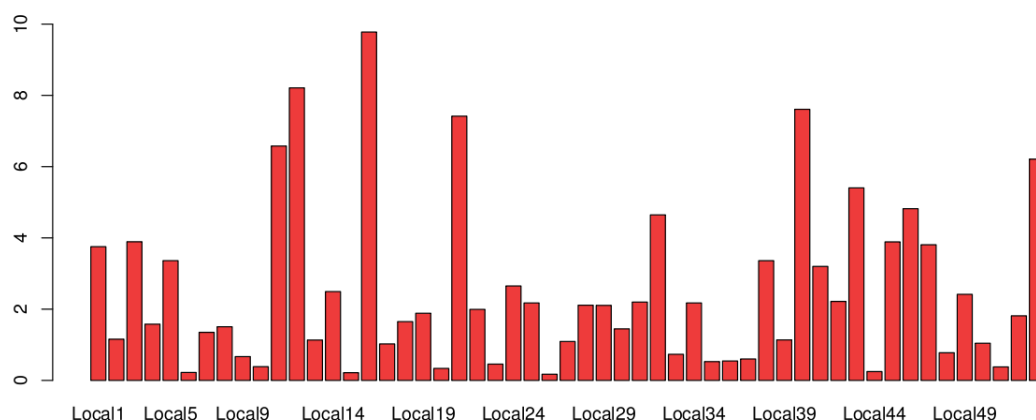


FIGURA 3.3: Demanda média diária por local

FONTE: O autor(2011)

As figuras 3.4 e 3.5 apresentam, respectivamente, a demanda diária e a distribuição de frequência da demanda do local de atendimento 39 (indicado na figura

3.3 ). A demanda média diária desse local é de 8 refeições por dia. Entretanto, analisando-se as figuras 3.4 e 3.5 é possível visualizar o quanto a demanda é variável. Essa variabilidade ocorre também com os demais locais de atendimento.

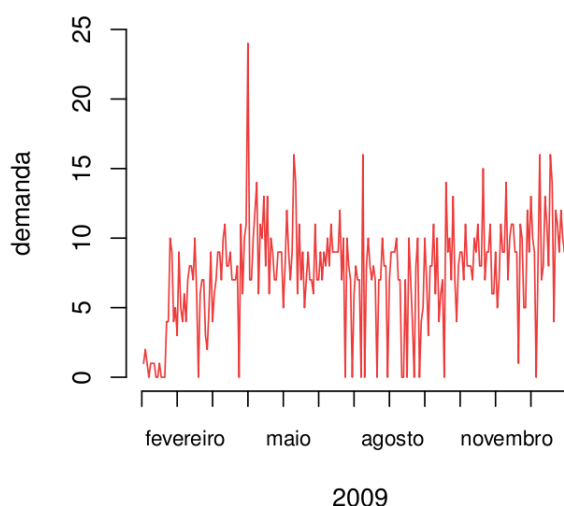


FIGURA 3.4: Demanda diária do local de atendimento 39

FONTE: O autor(2011)

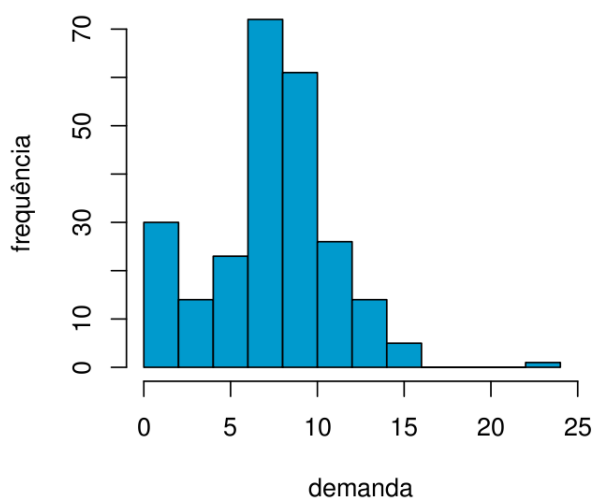


FIGURA 3.5: Distribuição de frequência de pedidos do local de atendimento 39

FONTE: O autor(2011)

Em média são entregues 130 refeições e atendidos 35 locais por dia. Entretanto, analisando-se os dados de entrega relativos ao ano de 2009, é possível perceber o quanto esses dados são aleatórios. Em alguns dias chegaram a ser entregues 230 refeições em aproximadamente 50 locais. Em outros dias apenas 15 locais foram

atendidos e por volta de 35 refeições entregues.

A ASSEMCO possui 2 veículos com capacidades iguais que partem do restaurante e possuem em torno de 2 horas para atender os locais de demanda. Atualmente existem 53 pontos de demanda cadastrados e a entrega de refeições se baseia na experiência dos funcionários encarregados do processo. Esses pontos de demanda estão distribuídos em duas rotas, os funcionários se apóiam em sua prática e conhecimento das condições viárias e de tráfego da região atendida para definir os roteiros. Entretanto, essa prática nem sempre é a mais eficaz porque nem todos os 53 locais precisam ser atendidos diariamente. Isso muitas vezes acarreta perda da qualidade da refeição que permanece muito tempo dentro do veículo e insatisfação dos clientes que não recebem o produto no horário determinado. A tabela 3.1 apresenta a configuração atual dos roteiros de entrega.

Rota	Qtde. de locais atendidos	Distância média percorrida (km)
1	26	60
2	27	75
Total	53	135

TABELA 3.1 : Distância média percorrida pelos veículos

FONTE: autor(2011)

Como os roteiros de entrega não são elaborados de maneira eficiente e não atendem as necessidades dos clientes, este trabalho visa auxiliar a ASSEMCO neste processo. Para isto o problema será abordado tanto de forma determinística quanto de forma estocástica.

Para o problema determinístico é apresentada uma proposta de solução em tempo real, que gere os roteiros de entrega no momento em que são conhecidos todos os pedidos, essa proposta é descrita no capítulo 4. Para o problema estocástico é apresentado, no capítulo 5, um estudo considerando a variabilidade do problema: demandas estocásticas e probabilidade de pedido dos clientes.

#### 4 RESOLUÇÃO DO PROBLEMA DE ENTREGA DE REFEIÇÕES - CASO DETERMINÍSTICO

Este capítulo descreve a forma de implementação da metodologia para gerenciamento de rotas da ASSEMCO. Para isso foi desenvolvido um *software*, denominado GERAROTA, utilizando a linguagem computacional Visual Basic 6.0.

Como citado no capítulo 3, os pedidos de refeições são, em sua maioria, feitos diariamente. Surge então a necessidade de uma ferramenta que elabore os roteiros de entrega, pois apesar dos lugares de entregas serem cadastrados, sua demanda só se torna conhecida no dia, e portanto, a mesma rota de entrega não precisa ser realizada todos os dias.

O primeiro passo para o desenvolvimento do *software* foi a preparação dos dados: relação dos lugares cadastrados com seus respectivos endereços e horários de entrega, e a distância real entre todos os lugares cadastrados, inclusive o depósito. Para obter as distâncias reais entre os pontos foi utilizada a *API do Google Maps*. Esses dados são armazenados em arquivos que são chamados durante a execução do programa.

O programa foi desenvolvido utilizando a estratégia Agrupar primeiro-Roteirizar depois, do inglês *Cluster first-Route second*. Essa estratégia consiste em:

**PASSO 1 - Clusterização:** Determinar os *clusters* que serão atendidos pelos veículos.

**PASSO 2- Roteirização:** Determinar o roteiro ótimo a ser percorrido em cada *cluster*.



## 4.1 CLUSTERIZAÇÃO

Nesta primeira etapa do programa são determinados os grupos de clientes que serão designados a cada um dos veículos.

### 4.1.1 Determinação das sementes

As sementes ou depósitos fictícios são pontos de atendimento ao redor dos quais serão agrupados os demais pontos. O número de sementes representa o número de veículos disponível no depósito. Esse número pode ser um número fornecido como um dado de entrada pelo usuário do programa, ou ainda pode ser determinado pelo próprio programa. O número mínimo de veículos necessário é dado por:

$$\text{Número de veículos} = \frac{\text{Demanda total}}{\text{Capacidade do veículo}}$$

Entretanto, esse número de veículos pode não ser suficiente quando são consideradas as janelas de tempo. Se isto acontecer, então mais um veículo é considerado.

Conhecido o número de veículos a serem utilizados, aplica-se o algoritmo das  $p$ -medianas de Teitz e Bart descrito na seção 2.3.1. Esse algoritmo é aplicado a fim de determinar no grafo formado pelos lugares cadastrados,  $p$ -pontos principais, que representam as sementes ou depósitos fictícios. O número  $p$  de medianas representa a quantidade de veículos disponíveis e a capacidade de cada mediana é dada pela capacidade do veículo. Os demais lugares serão designados a cada uma destas medianas de modo a formar os *clusters* (grupos de atendimento).

A figura 4.1 mostra a localização das sementes para uma dada simulação. Os pontos em amarelo e rosa indicam os locais ao redor dos quais os demais pontos serão agrupados.

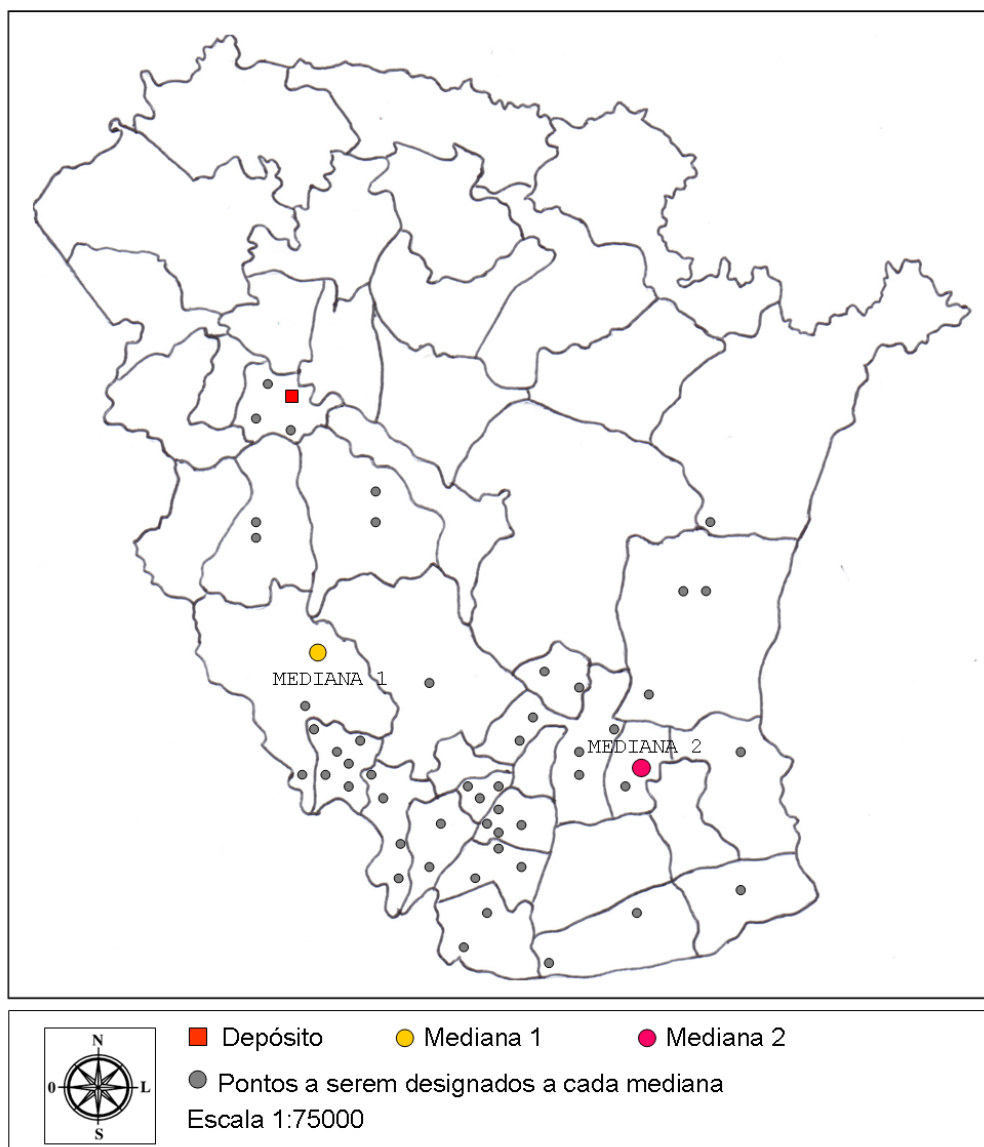


FIGURA 4.1: Localização das sementes para uma dada simulação

FONTE: O autor(2011)

#### 4.1.2 Determinação dos grupos de atendimento

Após terem sido definidas as medianas, aplica-se o algoritmo de designação de Gillett e Johnson, descrito na seção 2.4.1, a fim de designar os pontos de atendimento a cada uma das medianas encontradas, de maneira a formar os grupos de atendimento.

A figura 4.2 apresenta os pontos designados para cada mediana mostradas na figura 4.1. Os *clusters* são indicados pelos pontos em amarelo e rosa. Os demais

pontos representam locais onde não houve demanda.

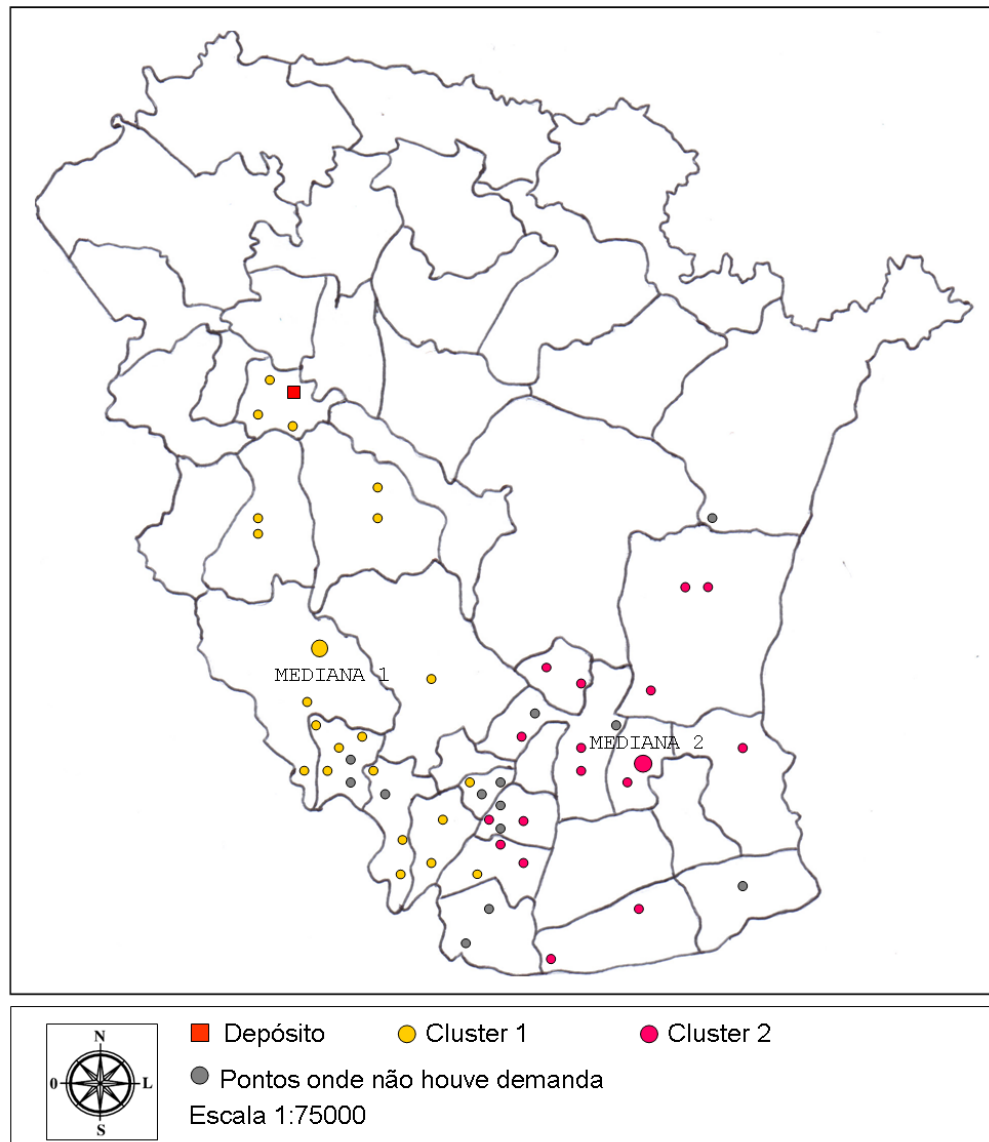


FIGURA 4.2: Determinação dos grupos de atendimento

FONTE: O autor(2011)

## 4.2 ROTEIRIZAÇÃO

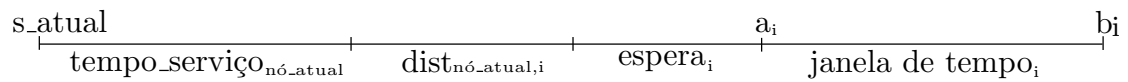
Obtidos os *clusters*, é necessário obter o roteiro dos veículos, ou seja, a sequência de clientes que cada veículo deverá atender. Para isso são utilizadas a heurística de inserção do vizinho mais próximo e a heurística de melhoramento 2-opt.

#### 4.2.1 Determinação do roteiro de entregas

O algoritmo do vizinho mais próximo, descrito na seção 2.1.1, foi adaptado de forma a considerar o horário em que cada cliente deve ser atendido (janela de tempo). Para isso foi incluída a seguinte modificação proposta por Pellegrini (2005): utilizar um custo ponderado ao invés somente da distância entre os pontos. Esse custo ponderado considera, além da distância entre os nós, a espera que poderá ocorrer caso se chegue a um nó antes do início de sua janela de tempo, bem como a urgência de se atender um nó, devido à proximidade do fim de sua janela.

A figura 4.3 representa duas situações: na primeira existe um tempo de espera, e na segunda não.

Situação 1:



Situação 2:

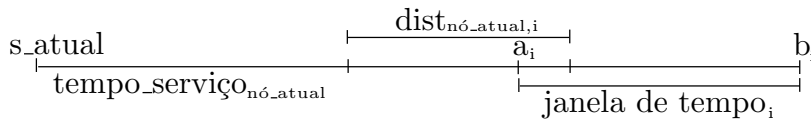


FIGURA 4.3: Intervalos de tempos considerados na heurística do vizinho mais próximo

FONTE: Santos(2006)

A janela de tempo para o cliente  $i$  é dada por  $[a_i, b_i]$ ,  $s\_atual$  representa o início do serviço no nó atual,  $tempo\_serviço_{nó\_atual}$  é o tempo necessário para atender o nó atual e  $dist_{nó\_atual,i}$  é a distância real entre o nó atual e o nó  $i$ . Dessa forma, a espera relacionada ao nó  $i$  é calculada da seguinte forma:

$$tempo\_viagem_{nó\_atual,i} = s\_atual + tempo\_serviço_{nó\_atual} + dist_{nó\_atual,i} \quad (4.1)$$

$$espera = \max\{a_i - tempo\_viagem_{nó\_atual,i}, 0\} \quad (4.2)$$

Se  $tempo\_viagem_{nó\_atual,i} < a_i$ , primeira situação indicada na figura 4.3, existe um tempo de espera e esse tempo é dado por  $a_i - tempo\_viagem_{nó\_atual,i}$ . Entretanto se  $tempo\_viagem_{nó\_atual,i} > a_i$  tem-se que  $(a_i - tempo\_viagem_{nó\_atual,i}) < 0$ , e portanto não há espera, como indicado na segunda situação na figura 4.3, logo  $espera = 0$ .

De maneira análoga, define-se a urgência relativa ao nó  $i$ :

$$urgencia = b_i - tempo\_viagem_{nó\_atual,i} \quad (4.3)$$

Pela equação (4.3) é possível perceber que enquanto  $urgencia$  assumir valores positivos é possível chegar ao nó antes do fechamento de sua janela. Dessa maneira, assegura-se que o cliente será atendido dentro do horário previsto.

Sendo assim, o custo ponderado é dado por:

$$custo = \alpha * distancia + \beta * espera + \gamma * urgencia \quad (4.4)$$

onde  $\alpha$ ,  $\beta$  e  $\gamma$  são parâmetros.

O algoritmo do vizinho mais próximo utilizará esse custo ponderado para construir a rota para o problema.

#### 4.2.2 Melhoria do roteiro encontrado

Gerado o roteiro pelo algoritmo do vizinho mais próximo é aplicada a heurística de melhoramento 2-opt, descrita na seção 2.1.1. No procedimento 2-opt também foi utilizado o custo ponderado dado pela equação 4.4 a fim de considerar as janelas de tempo dos clientes.

Como saída do programa tem-se uma lista ordenada dos lugares a serem atendidos, e o trajeto de ruas sugerido visualizado através do *site Google Maps*.

Na figura 4.4(a) é apresentado o roteiro de entregas a ser percorrido por um veículo em uma dada simulação. A imagem de satélite desse roteiro está apresentada na figura 4.4(b). A sequência do roteiro é [A B C D E F G H I J A], onde o ponto



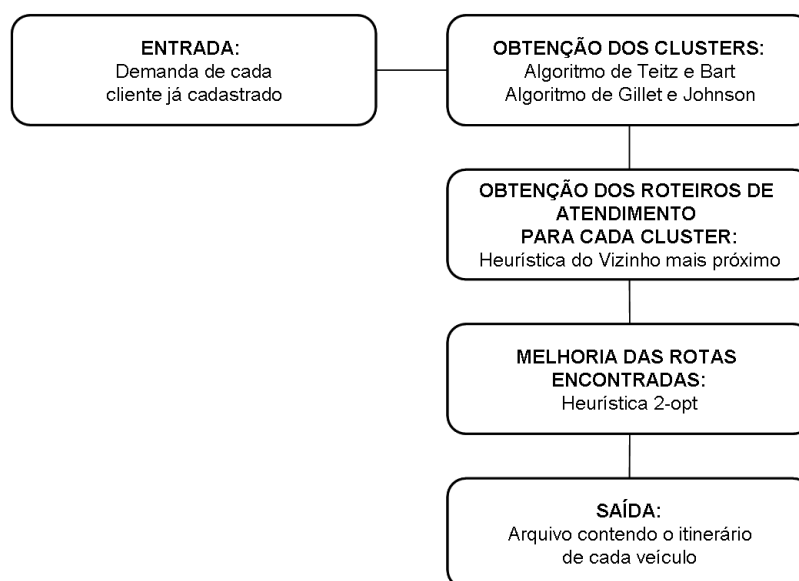


FIGURA 4.5: Fluxograma de funcionamento do programa GERAROTA

FONTE: O autor(2011)

### 4.3 RESULTADOS

Nesta seção são apresentados alguns resultados obtidos com a metodologia proposta.

As tabelas 4.1 e 4.2 apresentam uma comparação entre a distância total percorrida pelos veículos no trajeto atual utilizado pela ASSEMCO e nas simulações feitas usando o programa GERAROTA.

Na primeira simulação (TABELA 4.1) foi considerada a configuração de *clusters* já utilizada pela ASSEMCO e o programa foi usado apenas para traçar os roteiros de entrega. Na segunda simulação (TABELA 4.2) o programa gerou os *clusters* e as rotas de atendimento. Em todas as situações foram considerados todos os pontos de demanda para que os resultados pudessem ser comparados com o atual roteiro feito pela ASSEMCO.

A solução encontrado pela metodologia proposta, em ambos os casos, foi menor que o trajeto praticado pelos motoristas da ASSEMCO. Mesmo utilizando os *clusters*

TABELA 4.1: Distância percorrida pelos veículos no trajeto usado pela ASSEMCO e na simulação 1

Cluster	Qtde. de locais atendidos	Trajetos atual(km)	Simulação 1(km)	Melhoria (%)
1	26	60	55,2	8
2	27	75	60,3	20
TOTAL	53	135	115,5	14

FONTE: O autor(2011)

TABELA 4.2: Distância percorrida pelos veículos no trajeto usado pela ASSEMCO e na simulação 2

Cluster	Qtde. de locais atendidos		Distância percorrida(km)		Melhoria (%)
	Trajetos atual	Simulação 2	Trajetos atual	Simulação 2	
1	26	18	60	37,3	37
2	27	35	75	69,6	7
TOTAL	53	53	135	106,9	22

FONTE: O autor(2011)

pré-definidos a solução encontrada foi satisfatória.

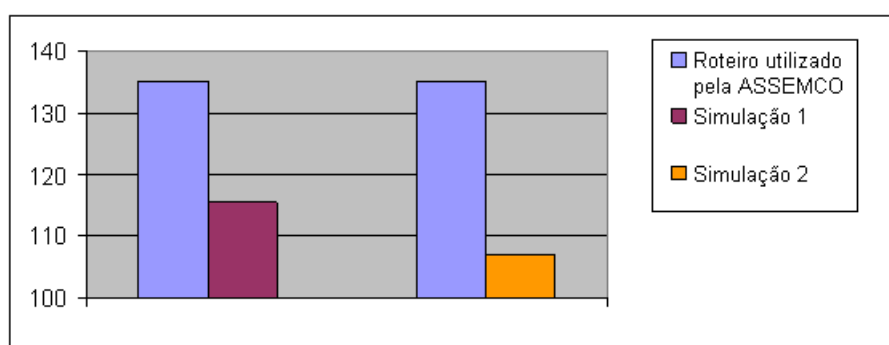


FIGURA 4.6: Comparação entre as rotas utilizadas pela ASSEMCO e encontradas pelo programa GERAROTA

FONTE: O autor(2011)

Foram executadas 75 simulações, o tempo de execução médio do programa GERAROTA é de aproximadamente 13 segundos. As soluções otimizadas pelo programa GERAROTA apresentaram uma redução média de 18% em relação aos roteiros normalmente utilizados pela ASSEMCO, o que representa em média 24,3 km diários.

Os roteiros encontrados com a metodologia proposta se mostraram satisfatórios em relação os roteiros normalmente praticados pela ASSEMCO. Ao minimizar a distância percorrida pelos veículos consequentemente obtém-se uma redução no tempo



de realização das entregas e de permanência da refeição dentro do veículo, o que sem dúvida é fundamental para conservação da qualidade da mesma. Outras medidas relativas ao acondicionamento das refeições também devem ser consideradas, mas não foram o foco deste trabalho.

Entretanto, alterar os roteiros de entrega todos os dias, muitas vezes acarreta na insatisfação do motorista encarregado do processo. Então, a fim de analisar o quanto a variabilidade da demanda e dos clientes afeta os roteiros de entrega, no capítulo 5 é apresentado um estudo estocástico do problema considerando o histórico de pedidos dos clientes durante o ano de 2009.

## 5 RESOLUÇÃO DO PROBLEMA DE ENTREGA DE REFEIÇÕES - CASO ESTOCÁSTICO

Este capítulo apresenta um modelo estocástico linear proposto para o problema de roteamento de veículos. Sua resolução é feita *a priori*, baseada na probabilidade de presença dos clientes, isto é, na probabilidade do cliente possuir demanda.

No PRVJT padrão as janelas de tempo devem ser respeitadas. No modelo proposto é permitido que o cliente seja atendido fora da janela de tempo prevista, neste caso é considerado um desvio da janela (por atraso ou antecedência) e uma penalidade é imposta. O objetivo do modelo é minimizar os desvios das janelas de tempo, além da distância percorrida pelos veículos e o número de veículos utilizados.

Foram incorporados ao modelo a probabilidade de presença dos clientes, que neste estudo de caso significa que o cliente fez um pedido, a probabilidade de utilização do arco  $(i, j)$ , calculada considerando-se a probabilidade de presença dos nós  $i$  e  $j$ , e dos nós intermediários. O tempo de viagem entre os clientes foi calculado baseado na probabilidade de presença destes.

O modelo proposto foi baseado em Azi, Gendreau e Potvin (2007), Bard (2010) e Mark e Guo (2004) e pode ser definido como segue.

Seja um grafo orientado  $G = (N, A)$ , onde  $N = V \cup \{0, n+1\}$ ,  $V = \{1, \dots, n\}$  é o conjunto de nós que representam os clientes, 0 e  $n+1$  são os nós que representam o depósito e  $A = \{(i, j) : i, j \in N, i \neq j\}$  o conjunto de arcos associados às conexões entre os nós. O conjunto de veículos é denotado por  $K$ , e cada veículo  $k \in K$  tem capacidade  $Q$ .

A cada arco  $(i, j) \in A$  estão associados um custo (distância)  $c_{ij}$ , um tempo de viagem  $t_{ij}$  e uma probabilidade  $\bar{p}_{ij}$  de presença na rota. Cada cliente  $i \in V$  possui uma demanda  $d_i$ , uma probabilidade de presença  $p_i$ , uma janela de tempo  $[a_i, b_i]$  e uma

penalidade  $\lambda_i$  caso a janela de tempo prevista não seja respeitada. O custo de utilizar um veículo é dado por  $c_v$  e é igual para todos os veículos.

As variáveis utilizadas no modelo são definidas como:

$$x_{ij}^k = \begin{cases} 1 & \text{se o veículo } k \text{ percorre o arco } (i, j), \forall k \in K \text{ e } \forall (i, j) \in A \\ 0 & \text{caso contrário} \end{cases}$$

$$y_i^k = \begin{cases} 1 & \text{se o cliente } i \text{ é atendido pelo veículo } k, \forall k \in K \text{ e } \forall i \in V \\ 0 & \text{caso contrário} \end{cases}$$

$s_i^k$  = instante em que o veículo  $k$  começa a servir o cliente  $i$ ,  $\forall k \in K \text{ e } \forall i \in V$ .

$D_i^+$  = desvio da janela de tempo relativa ao cliente  $i$ , caso este seja atendido com atraso.

$D_i^-$  = desvio da janela de tempo relativa ao cliente  $i$ , caso este seja atendido com antecedência.

Considerando  $M$  uma constante qualquer grande o suficiente, tem-se:

### **MODELO-PRVER: Problema de roteamento de veículos na entrega de refeições**

$$\text{minimizar} \quad \sum_{i \in V} \lambda_i (D_i^+ + D_i^-) - \sum_{k \in K} \sum_{(i,j) \in A} c_{ij} \bar{p}_{ij} x_{ij}^k + \sum_{k \in K} \sum_{i \in V} c_v y_i^k \quad (5.1)$$

sujeito a

$$\sum_{j \in N} x_{ij}^k = y_i^k \quad i \in V, k \in K \quad (5.2)$$

$$\sum_{k \in K} y_i^k = 1 \quad i \in V \quad (5.3)$$

$$\sum_{i \in N} x_{ih}^k - \sum_{j \in N} x_{hj}^k = 0 \quad h \in N, k \in K \quad (5.4)$$

$$\sum_{i \in N} x_{0i}^k = 1 \quad k \in K \quad (5.5)$$

$$\sum_{i \in N} x_{i(n+1)}^k = 1 \quad k \in K \quad (5.6)$$

$$\sum_{i \in N} d_i y_i^k \leq Q \quad k \in K \quad (5.7)$$

$$\sum_{i \in N} \sum_{j \in N} t_{ij} x_{ij}^k \leq T \quad k \in K \quad (5.8)$$

$$s_i^k + t_{ij} - M(1 - x_{ij}^k) \leq s_j^k \quad (i, j) \in A, k \in K \quad (5.9)$$

$$-D_i^- + a_i y_i^k \leq s_i^k \leq b_i y_i^k + D_i^+ \quad i \in N, k \in K \quad (5.10)$$

$$s_i^k \leq M y_i^k \quad i \in V, k \in K \quad (5.11)$$

$$x_{ij}^k \in \{0, 1\} \quad i, j \in N, k \in K \quad (5.12)$$

$$y_i^k \in \{0, 1\} \quad i \in N, k \in K \quad (5.13)$$

$$s_i^k \geq 0 \quad i \in N, k \in K \quad (5.14)$$

$$D_i^-, D_i^+ \geq 0 \quad i \in N \quad (5.15)$$

A função objetivo (5.1) minimiza uma ponderação entre os desvios de janela, a distância percorrida pelos veículos e o número de veículos utilizados. A distância percorrida é calculada através do somatório da distância percorrida por cada veículo da solução multiplicada pela probabilidade de presença do trajeto percorrido. O valor final da solução representa um limite inferior para o custo total esperado da rota.

As restrições (5.2) e (5.3) asseguram que cada cliente seja visitado somente uma vez. As restrições (5.4), (5.5) e (5.6) são relativas à estrutura de redes e exigem que cada veículo parta do depósito e retorne ao mesmo somente uma vez, e deixe um determinado nó somente se entrar nele. As restrições (5.7) e (5.8) impõem que cada rota não exceda a capacidade  $Q$  do veículo e o limite máximo de tempo  $T$ , respectivamente. As restrições (5.9) garantem que o atendimento ao cliente  $i$  ocorra antes do atendimento ao cliente  $j$ , quando o cliente  $j$  for o sucessor do cliente  $i$  na rota percorrida pelo veículo  $k$ . As restrições (5.10) delimitam as janelas de tempo. As restrições (5.11) associam o instante de atendimento do cliente  $i$  ao veículo  $k$ . As restrições (5.12), (5.13), (5.14) e (5.15) indicam o tipo de variável.

A probabilidade de presença  $\bar{p}_{ij}$  do arco  $(i, j)$  é calculada segundo Mark e Guo (2004):

$$\bar{p}_{ij} = p_i p_j \prod_{k=i+1}^{j-1} (1 - p_k), \quad i, j \in N \quad (5.16)$$

Para ilustrar o cálculo considere a configuração de arcos dada na figura 5.1, onde os valores sobre os arcos indicam os custos associados a cada arco:

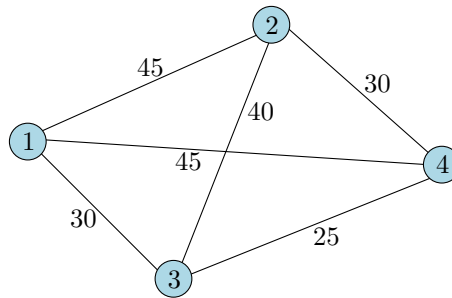


FIGURA 5.1: Exemplo de configuração de arcos

FONTE: O autor(2011)

Suponha que se queira ir do nó 1 para o nó 4. Sejam  $p_1 = p_4 = 0,8$ ,  $p_2 = 0,5$  e  $p_3 = 0,20$ , onde  $p_i$  é a probabilidade do cliente  $i$  fazer um pedido,  $i = 1, 2, 3, 4$ . A probabilidade de utilização do arco (1,4) é:

$$\bar{p}_{14} = p_1 p_4 \prod_{k=2}^3 (1 - p_k) \quad (5.17)$$

$$\bar{p}_{14} = p_1 p_4 (1 - p_2)(1 - p_3) \quad (5.18)$$

$$\bar{p}_{14} = 0,8 \cdot 0,8 \cdot (1 - 0,50)(1 - 0,20) \quad (5.19)$$

$$\bar{p}_{14} = 0,256 \quad (5.20)$$

Agora suponha  $p_2 = p_3 = 0,9$ , onde  $p_2$  e  $p_3$  indicam as probabilidades dos clientes 2 e 3 fazerem pedido. Logo,

$$\bar{p}_{14} = 0,8 \cdot 0,8 \cdot (1 - 0,9)(1 - 0,9) \quad (5.21)$$

$$\bar{p}_{14} = 0,0064 \quad (5.22)$$

Assim, quanto menor a probabilidade de presença dos nós intermediários, nós 2 e 3, maior será a probabilidade de se utilizar o arco que liga os nós diretamente, ou seja, arco (1,4).

O tempo de viagem  $t_{ij}$  é calculado segundo Bard (2010). O referido autor desenvolveu um esquema de pesos que considera as  $m_i$  localizações mais próximas do cliente  $i$ . Também foi utilizado um ranking  $R_{i,[j]}$  dos  $j^{th}$  clientes mais próximos de  $i$ , onde  $R_{i,[j]} = k$ , para  $k = 1, \dots, m_i$ , sendo  $k$  a classificação de proximidade de  $j$  em relação ao nó  $i$ .

Por exemplo, considere o grafo apresentado na figura 5.1. Supondo  $m_4 = 2$ , tem-se que os dois nós mais próximos ao nó 4 são os nós  $\{3, 2\}$  respectivamente. Logo, o ranking das duas localizações mais próximas do nó 4, é dado por:

$$R_{4,[3]} = 1 \quad (\text{primeiro nó mais próximo}) \quad (5.23)$$

$$R_{4,[2]} = 2 \quad (\text{segundo nó mais próximo}) \quad (5.24)$$

Seja  $p_j$  a probabilidade do cliente  $j$  requerer a visita num determinado dia, e seja  $w_{i,[j]}$  a probabilidade do cliente  $[j]$  seguir o cliente  $i$  na rota. A equação que calcula o valor do peso  $w_{i,[j]}$  é dada pela equação (5.25):

$$w_{i,[j]} = \frac{p_j (m_i - R_{i,[j]} + 1)}{\sum_{k=1}^{m_i} p_k (m_i - R_{i,[k]} + 1)}, \quad i, j \in N \quad (5.25)$$

Sendo  $c(i, [j])$  a distância entre os clientes  $i$  e  $j$ , e  $Vel(i, [j])$  a velocidade média do veículo, o tempo de viagem  $t_{i[j]}$  é dado por:

$$t_{i[j]} = w_{i,[j]} \frac{c(i, [j])}{Vel(i, [j])}, \quad i, j \in N \quad (5.26)$$

Segundo Bard (2010), o objetivo da equação (5.26) é considerar tanto a distância entre os clientes  $i$  e  $j$  quanto a probabilidade da visita. Na equação (5.25) o denomi-

nador normaliza os valores  $w_{i,[j]}$ , de forma que a soma deles seja 1, gerando assim uma probabilidade de visita aproximada.

Como exemplo, considere novamente o grafo da figura 5.1, com as probabilidades  $p_1 = p_4 = 0,8$ ,  $p_2 = 0,5$  e  $p_3 = 0,20$  associadas aos nós 1, 4, 2 e 3 respectivamente. Suponha que se queira calcular o tempo de viagem do nó  $i = 4$  para os nós  $i = 2, 3$ .

Conforme apresentado em (5.24) e (5.23):  $R_{4,[3]} = 1$  e  $R_{4,[2]} = 2$ . Calculando-se o valor dos pesos através da equação (5.25), tem-se que:

$$w_{4,[2]} = \frac{p_2 (m_4 - R_{4,[2]} + 1)}{\sum_{k=1}^{m_4} p_k (m_4 - R_{4,[k]} + 1)} \quad (5.27)$$

$$= \frac{0,50(2 - 2 + 1)}{0,20(2 - 1 + 1) + 0,50(2 - 2 + 1)} = 0,56 \quad (5.28)$$

$$w_{4,[3]} = \frac{p_3 (m_4 - R_{4,[3]} + 1)}{\sum_{k=1}^{m_4} p_k (m_4 - R_{4,[k]} + 1)} \quad (5.29)$$

$$= \frac{0,20(2 - 1 + 1)}{0,20(2 - 1 + 1) + 0,50(2 - 2 + 1)} = 0,44 \quad (5.30)$$

Conhecidos os valores dos pesos, é possível calcular o tempo de viagem através da equação (5.26). As distâncias entre os arcos estão indicadas na figura 5.1.

Considerando-se a velocidade dos veículos constante  $Vel = 50km/h = 13,9m/s$ , tem-se:

$$t_{4[2]} = w_{4,[2]} \frac{c(4, [2])}{MPH} = 0,56 \frac{30}{13,9} = 1,21s \quad (5.31)$$

$$t_{4[3]} = w_{4,[3]} \frac{c(4, [3])}{MPH} = 0,44 \frac{25}{13,9} = 0,79s \quad (5.32)$$

## 5.1 IMPLEMENTAÇÃO DO ANT COLONY SYSTEM PARA RESOLUÇÃO DO MODELO PRVER

A meta-heurística ACS, descrita na seção 2.6.2, foi implementada para resolver o modelo PRVER proposto no início deste capítulo, visto que, em virtude da complexidade computacional deste, resolvê-lo de forma exata torna-se inviável para grandes instâncias.

O algoritmo *Ant Colony System* foi implementado em Matlab e foi baseado no trabalho de Bullnheimer, Hartl e Strauss (1999), que aplicaram o AS ao problema de roteamento de veículos capacitado.

Foram utilizadas  $m = n$  formigas, onde  $n$  indica o número de nós. A cada formiga está associada uma variável chamada  $h_{atual,i}$  que indica o momento em que a formiga chega ao nó  $i$ . Inicialmente todas as  $n$  formigas estão localizados no nó 0 e  $h_{atual,i} = 0$ .

Para seleccionar o próximo nó a ser visitado as formigas utilizam a regra de decisão dada por (2.30). O processo se repete até que todos os nós tenham sido visitados. Entretanto, se a escolha de um nó implicar em uma solução infactível, ou seja, se a capacidade do veículo for excedida, o depósito é escolhido e uma nova rota é iniciada.

Cada vez que a formiga move-se do nó  $i$  para o nó  $j$ , a variável  $h_{atual,i}$  é atualizada da seguinte maneira:

$$h_{atual,j} = h_{atual,i} + t_{ij} \quad (5.33)$$

Onde  $t_{ij}$  é o tempo de viagem entre os nós  $i$  e  $j$ .

Durante a construção das rotas as formigas liberam feromônio que fica no caminho percorrido por elas. As atualizações local e global do feromônio são calculadas de acordo com as expressões (2.31) e (2.32), respectivamente. Sendo que a atualização global privilegia os arcos pertencentes à melhor solução global, ou seja, a solução de menor custo total.



Sendo  $[a_i, b_i]$  a janela de tempo associada ao cliente  $i$ , os desvios de janela são dados por:

$$D_i^- = a_i - h_{atual,i} \quad (5.34)$$

$$D_i^+ = h_{atual,i} - b_i \quad (5.35)$$

O custo total de cada rota é calculado da seguinte forma:

$$c_{total} = \sum_{i=1}^n \lambda_i (D_i^- + D_i^+) + \sum_{(i,j) \in S} c_{ij} \quad (5.36)$$

Onde:

$S$  é a rota construída pela formiga.

$c_{ij}$  é o custo associado ao arco  $(i, j)$ .

$\lambda_i$  é a penalidade associada ao cliente  $i$ , caso ele seja atendido fora de sua janela de tempo.

Dessa forma, quanto menores forem os desvios de janela menor será o custo total da rota.

## 5.2 RESULTADOS

Os dados para a resolução do modelo aplicado ao problema da ASSEMCO foram calculados com base nas refeições entregues no período de janeiro a dezembro de 2009.

A probabilidade  $p_i$  do cliente requerer a visita num determinado dia foi calculada pelo número de ocorrências de pedido no ano sobre o total de dias em que houve entrega.

A demanda  $d_i$  de cada cliente é dada por um número aleatório obtido de uma distribuição de probabilidade de acordo com os cenários:

**CENÁRIO 1:** Distribuição de Poisson com parâmetro  $\Theta$  igual à demanda média anual do cliente.

**CENÁRIO 2:** Distribuição Normal com parâmetros  $\mu$  e  $\sigma$  iguais à demanda média anual e desvio padrão do cliente, respectivamente.

**CENÁRIO 3:** Distribuição Gama com parâmetros  $\alpha$  igual à demanda média anual do cliente e  $\beta = 1$ .

**CENÁRIO 4:** Distribuição Uniforme no intervalo  $[a, b]$  com  $a$  e  $b$  iguais ao valor mínimo e máximo de pedidos num dia, ocorridos durante o ano.

Essas distribuições estão detalhadas no anexo A.

A janela de tempo de cada cliente é gerada através de uma Distribuição Uniforme no intervalo  $[20, 100]$ . O número de veículos é variável em função da demanda, a capacidade dos veículos é de 100 unidades e a velocidade média é considerada constante de 60 km/h.

As penalidades variam de acordo com o nó considerado. Entretanto, nós que têm janelas de tempo mais rígidas possuem maior valor de penalidade.

Foram criadas rotinas em *Matlab* para gerar as matrizes de probabilidade de presença dos arcos e de tempos de viagem, dadas pelas equações (5.16) e (5.26), respectivamente.

Esse modelo foi resolvido de forma exata e também utilizando a meta-heurística ACS. Para testar a qualidade da solução heurística, supõe-se que quando o problema não permitir solução exata o comportamento seja análogo.

#### 5.2.1 Método exato aplicado ao modelo PRVER

O modelo foi resolvido de maneira exata utilizando o pacote computacional *Lingo 9.0*, executado em um computador com processador Intel Core Duo T6400, 2.20 GHz, 2GB de RAM e sistema operacional Windows 7. Este pacote computacional utiliza

a técnica *branch-and-bound* para resolução de problemas de programação inteira ou mista.

Devido à complexidade computacional do modelo proposto, este só pode ser resolvido exatamente para pequenas instâncias. Foram considerados  $n = 10, 15$  e 20 nós.

Nos resultados apresentados a seguir o nó  $n = 1$  representa o depósito,  $s_i$  representa o instante de início de atendimento ao cliente  $i$ ,  $D_i^-$  e  $D_i^+$  são os desvios de janela relativos ao cliente  $i$  e  $[a_i, b_i]$  são as janelas de tempo consideradas. A primeira coluna das tabelas indicam a sequência de atendimento.

As tabelas 5.1, 5.2, 5.3 e 5.4 apresentam os valores encontrados para os instantes de início de atendimento, os desvios de janela e as janelas de tempo consideradas para  $n = 10$  nós (9 clientes e o depósito). As demandas são dadas pelos cenários 1, 2, 3 e 4 respectivamente. As linhas em destaque indicam os clientes para os quais houve desvio da janela de tempo.

TABELA 5.1: Solução exata do modelo para 10 nós - CENÁRIO 1

Nó	$s_i$	$D_i^-$	$D_i^+$	$a_i$	$b_i$
1	0,00	0,00	0,00	0	100
6	6,40	33,60	0,00	40	74
10	17,08	24,92	0,00	42	94
5	26,07	14,93	0,00	41	65
8	34,13	5,87	0,00	40	67
4	43,08	5,92	0,00	49	57
7	55,40	0,00	0,00	48	63
3	61,37	0,00	0,00	28	91
2	68,11	0,00	0,00	50	85
9	77,81	0,00	3,81	35	74
1	86,43	0,00	0,00	0	100

FONTE: O autor(2011)

TABELA 5.2: Solução exata do modelo para 10 nós - CENÁRIO 2

Nó	$s_i$	$D_i^-$	$D_i^+$	$a_i$	$b_i$
1	0,00	0,00	0,00	0	100
6	6,40	28,60	0,00	35	55
10	17,08	25,92	0,00	43	52
5	26,07	2,93	0,00	29	85
8	34,13	0,00	0,00	31	57
4	43,08	0,00	0,00	34	51
7	55,40	0,00	0,00	26	99
3	61,37	0,00	0,00	30	97
2	68,11	0,00	0,00	45	99
9	77,81	0,00	0,00	21	83
1	86,43	0,00	0,00	0	100

FONTE: O autor(2011)

TABELA 5.3: Solução exata do modelo para 10 nós - CENÁRIO 3

Nó	$s_i$	$D_i^-$	$D_i^+$	$a_i$	$b_i$
1	0,00	0,00	0,00	0	100
9	8,62	12,38	0,00	21	83
4	20,96	7,04	0,00	28	91
7	33,27	0,00	0,00	26	99
2	43,05	6,95	0,00	50	78
3	49,78	0,22	0,00	50	85
8	54,34	0,00	0,00	31	57
5	62,40	0,00	5,40	49	57
10	71,39	0,00	19,39	43	52
6	82,07	0,00	17,07	41	65
1	89,01	0,00	0,00	0	100

FONTE: O autor(2011)

TABELA 5.4: Solução exata do modelo para 10 nós - CENÁRIO 4

Nó	$s_i$	$D_i^-$	$D_i^+$	$a_i$	$b_i$
1	0,00	0,00	0,00	0	100
6	6,40	20,60	0,00	27	48
10	17,08	3,92	0,00	21	46
4	26,99	0,01	0,00	27	31
9	39,32	0,00	0,00	22	42
3	45,25	0,00	0,00	24	54
7	51,23	0,00	0,00	28	55
2	61,01	0,00	0,00	27	68
8	68,46	0,00	3,46	23	65
5	76,52	0,00	10,52	30	66
1	83,35	0,00	0,00	0	100

FONTE: O autor(2011)

Analisando-se as tabelas 5.1 a 5.4 observa-se que os desvios de janela foram minimizados. Observa-se, como esperado, que os clientes no início da rota apresentam desvio de janela por antecedência no atendimento ( $D_i^-$ ), enquanto que clientes atendidos no final da rota apresentam desvio por atraso ( $D_i^+$ ).

A figura 5.2 apresenta a configuração da solução exata do modelo para 10 nós.

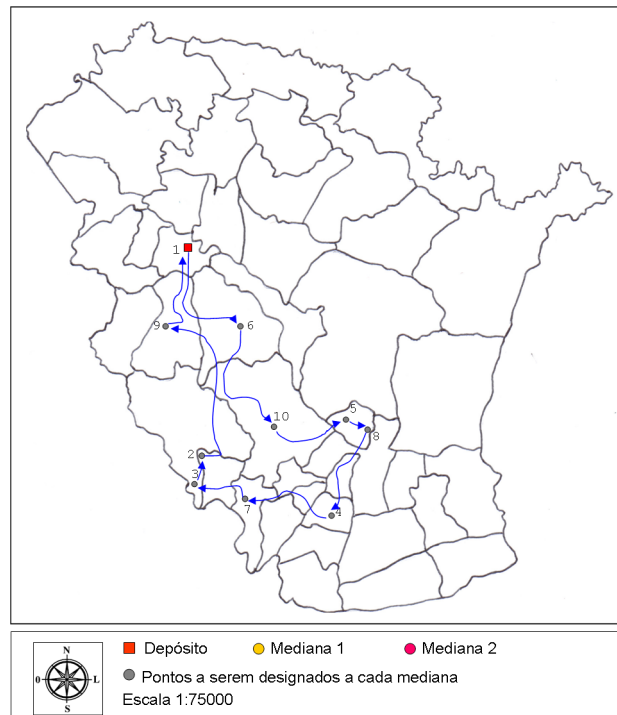


FIGURA 5.2: Solução exata do modelo para 10 nós - CENÁRIO 1

FONTE: O autor(2011)

Um resumo das soluções encontradas para as instâncias testadas e o tempo de execução das mesmas é apresentado na tabela 5.5. Percebe-se que, embora apenas 3 clientes tenham sido atendidos fora de sua janela de tempo na configuração apresentada na tabela 5.2, o menor custo foi dado pela configuração apresentada na tabela 5.4, onde 5 clientes foram atendidos com desvio. Isso ocorre pois o custo total da solução varia não só em função da distância percorrida pelos veículos, mas também em função dos desvios de janela.

TABELA 5.5: Resumo das soluções encontradas para  $n = 10$

Cenário	Custo total	Veículos utilizados	Cientes atendidos fora da janela	Tempo de execução (s)
1	260,32	1	6	32
2	220,14	1	3	30
3	243,19	1	7	28
4	206,41	1	6	31

FONTE: O autor(2011)

Os resultados apresentados nas tabelas 5.6 e 5.7 ilustram a influência da penalidade  $\lambda$  na construção das rotas foram executadas simulações utilizando diferentes valores de penalidade para os clientes. Foi considerada a mesma configuração de demanda e janela de tempo utilizada na solução dada na tabela 5.3.

TABELA 5.6: Solução para valores de penalidade entre  $[0, 10]$

Nó	$s_i$	$D_i^-$	$D_i^+$	$\lambda_i$	$a_i$	$b_i$
1	100,00	0,00	0,00	0,00	0	100
9	8,62	34,38	0,00	1,25	21	83
2	18,33	2,67	0,00	1,25	50	78
3	25,06	2,94	0,00	1,25	50	85
7	31,04	0,00	0,00	3,50	26	99
4	43,35	0,00	0,00	10,00	28	91
8	52,31	0,00	0,00	1,25	31	57
5	60,37	0,00	0,00	1,25	49	57
10	69,36	0,00	4,36	3,50	43	52
6	80,03	0,00	0,00	3,50	41	65
1	100,00	0,00	0,00	0,00	0	100

FONTE: O autor(2011)

TABELA 5.7: Solução para valores de penalidade entre  $[0,6]$ 

Nó	$s_i$	$D_i^-$	$D_i^+$	$\lambda_i$	$a_i$	$b_i$
1	100,00	0,00	0,00	0,00	0	100
7	8,48	22,52	0,00	1,25	26	99
4	20,80	5,20	0,00	4,00	28	91
9	33,13	9,87	0,00	2,00	21	83
3	39,06	0,00	0,00	1,25	50	85
2	45,80	0,00	0,00	4,00	50	78
8	53,25	0,00	0,00	5,75	31	57
5	61,31	0,00	0,00	1,25	49	57
10	70,30	0,00	5,30	1,25	43	52
6	80,98	0,00	0,00	4,00	41	65
1	100,00	0,00	0,00	0,00	0	100

FONTE: O autor(2011)

Analisando-se as tabelas 5.3, 5.6 e 5.7 observa-se que as rotas variam com a mudança de penalidade, visto que os clientes associados a uma penalidade maior têm prioridade no atendimento.

Os resultados para  $n = 15$  nós, considerando-se os cenários 1, 2, 3 e 4, estão apresentados nas tabelas 5.8, 5.9, 5.10 e 5.11.

TABELA 5.8: Solução exata do modelo para 15 nós - CENÁRIO 1

Nó	$s_i$	$D_i^-$	$D_i^+$	$a_i$	$b_i$
1	0,00	0,00	0,00	0	100
13	7,11	37,89	0,00	45	85
4	16,73	16,27	0,00	33	72
12	26,99	1,01	0,00	28	91
5	36,27	4,73	0,00	41	65
11	45,73	0,00	0,00	40	67
7	46,93	0,00	0,00	42	94
9	48,30	0,00	0,00	35	74
3	54,23	0,00	0,00	49	57
8	58,79	0,00	0,00	48	63
2	66,24	0,00	4,24	42	62
10	74,38	0,00	0,00	50	85
14	77,61	0,00	3,61	40	74
6	89,77	0,00	23,77	41	66
15	101,66	0,00	6,66	49	95
1	108,98	0,00	0,00	0	100

FONTE: O autor(2011)

TABELA 5.9: Solução exata do modelo para 15 nós - CENÁRIO 2

Nó	$s_i$	$D_i^-$	$D_i^+$	$a_i$	$b_i$
1	0,00	0,00	0,00	0	100
12	7,45	25,55	0,00	33	72
4	17,71	31,29	0,00	49	57
8	26,67	13,33	0,00	40	67
5	34,73	6,27	0,00	41	65
10	43,72	0,00	0,00	42	62
2	51,85	0,00	0,00	50	85
9	61,55	0,00	0,00	35	74
7	62,91	0,00	0,00	48	63
11	64,12	0,00	0,00	42	94
14	65,25	0,00	0,00	41	66
6	77,40	0,00	3,40	40	74
15	89,29	0,00	4,29	45	85
3	94,47	0,00	3,47	28	91
13	99,40	0,00	4,40	49	95
1	106,52	0,00	0,00	0	100

FONTE: O autor(2011)

Onde  $s_i$  representa o instante de início de atendimento ao cliente  $i$ ,  $D_i^-$  e  $D_i^+$  são os desvios de janela relativos ao cliente  $i$  e  $[a_i, b_i]$  são as janelas de tempo consideradas.

TABELA 5.10: Solução exata do modelo para 15 nós - CENÁRIO 3

Nó	$s_i$	$D_i^-$	$D_i^+$	$a_i$	$b_i$
1	0,00	0,00	0,00	0	100
10	7,14	14,86	0,00	21	31
5	16,13	10,87	0,00	27	31
15	26,00	0,00	0,00	22	42
11	27,10	0,00	0,00	30	38
14	28,22	0,00	0,00	23	65
7	29,78	0,00	0,00	27	48
9	31,14	0,00	0,00	21	46
2	40,85	0,00	0,00	26	60
8	48,30	0,00	0,00	28	55
3	52,86	0,00	0,00	27	68
12	58,05	0,00	0,00	24	68
4	68,31	0,00	14,31	24	54
13	77,93	0,00	36,93	29	41
6	88,31	0,00	22,31	30	66
1	94,71	0,00	0,00	0	100

FONTE: O autor(2011)

TABELA 5.11: Solução exata do modelo para 15 nós - CENÁRIO 4

Nó	$s_i$	$D_i^-$	$D_i^+$	$a_i$	$b_i$
1	0,00	0,00	0,00	0	100
6	6,40	21,60	0,00	28	51
13	16,78	0,00	0,00	12	36
4	26,40	0,00	0,00	26	36
15	36,66	0,00	0,00	31	58
11	37,76	0,00	0,00	19	69
14	38,88	0,00	0,00	25	62
7	40,44	0,00	0,00	28	57
9	41,80	0,00	0,00	13	46
3	47,74	0,00	0,00	19	56
10	52,71	0,00	3,71	13	49
2	60,84	0,00	0,00	27	68
8	68,30	0,00	3,30	18	65
5	76,36	0,00	9,36	35	67
12	85,63	0,00	40,63	31	45
1	93,08	0,00	0,00	0	100

FONTE: O autor(2011)

Observando-se as tabelas 5.8, 5.9, 5.10 e 5.11 é possível perceber que nas quatro simulações os clientes 5 e 6 foram atendidos com desvio, indicando que a penalidade associada a esses clientes é baixa, visto que estes não são tão exigentes em relação ao cumprimento do horário previsto para a entrega.

A tabela 5.12 apresenta um resumo das soluções encontradas para  $n = 15$ . Observa-se o aumento significativo no tempo de execução em relação ao número de nós. Para 10 nós foram gastos, em média, 30,2 segundos para resolução do modelo, enquanto que para 15 nós essa média foi de 16200 segundos, aproximadamente 4,5 horas.

TABELA 5.12: Resumo das soluções encontradas para  $n = 15$ 

Cenário	Custo total	Veículos utilizados	Clientes atendidos fora da janela	Tempo de execução (s)
1	225,92	1	8	14450
2	213,84	1	8	14423
3	237,12	1	5	18010
4	216,98	1	5	18025

FONTE: O autor(2011)

Para  $n = 20$  nós apenas uma instância foi testada, em virtude da complexidade computacional do modelo. Essa solução está apresentada na tabela 5.13. O tempo

de execução desta instância foi superior a 12 horas.

TABELA 5.13: Solução exata do modelo para 20 nós - CENÁRIO 4

Nó	$s_i$	$D_i^-$	$D_i^+$	$a_i$	$b_i$
1	0,00	0,00	0,00	0	100
6	6,40	2,60	0,00	9	32
15	18,29	0,00	0,00	15	32
19	24,86	0,00	0,00	7	69
10	28,22	0,00	0,00	21	81
20	32,09	0,00	0,00	25	65
3	37,99	0,00	0,00	16	88
17	44,18	0,00	0,00	3	53
11	45,49	0,00	0,00	3	59
14	46,61	0,00	0,00	23	79
7	48,17	0,00	0,17	23	48
13	49,37	0,00	11,37	19	38
4	58,99	0,00	1,99	21	57
12	69,25	0,00	9,25	13	60
5	78,53	0,00	0,00	30	89
18	90,05	0,00	4,05	1	86
8	97,75	0,00	12,75	7	85
16	103,52	0,00	62,52	7	41
2	111,16	0,00	53,16	16	58
9	120,86	0,00	39,86	14	81
1	129,48	0,00	0,00	0	100

FONTE: O autor(2011)

### 5.2.2 Meta-heurística *Ant Colony System* aplicada ao modelo PRVER

Para a implementação da meta-heurística ACS foram utilizadas  $m = n$  formigas conforme descrito na seção 5.1. Como critério de parada considerou-se não haver melhoria na solução durante 50 iterações ou se o número máximo de iterações foi atingido(250 iterações).

A concentração inicial de feromônio considerada foi  $\tau_0 = 0,0001$ , coeficiente de evaporação  $\rho = 0,25$ ,  $\alpha = 0,1$ ,  $\beta = 0,9$  e  $q_0 = 0,8$ .

Nas tabelas 5.14 e 5.15 encontram-se as soluções para  $n = 10$  e 15 nós, respectivamente.



TABELA 5.14: Solução ACS do modelo para 10 nós - CENÁRIO 1

Nó	$s_i$	$D_i^-$	$D_i^+$	$a_i$	$b_i$
1	0,00	0,00	0,00	0	100
6	6,40	33,60	0,00	40	74
10	17,08	24,92	0,00	42	94
5	26,07	14,93	0,00	41	65
8	34,13	5,87	0,00	40	67
4	43,08	5,92	0,00	49	57
7	55,40	0,00	0,00	48	63
3	61,37	0,00	0,00	28	91
2	68,11	0,00	0,00	50	85
9	77,81	0,00	3,81	35	74
1	86,43	0,00	0,00	0	100

FONTE: O autor(2011)

TABELA 5.15: Solução ACS do modelo para 15 nós - CENÁRIO 2

Nó	$s_i$	$D_i^-$	$D_i^+$	$a_i$	$b_i$
1	0,00	0,00	0,00	0	100
12	7,45	25,55	0,00	33	72
4	17,71	31,29	0,00	49	57
8	26,67	13,33	0,00	40	67
5	34,73	6,27	0,00	41	65
10	43,72	0,00	0,00	42	62
2	51,85	0,00	0,00	50	85
9	61,55	0,00	0,00	35	74
7	62,91	0,00	0,00	48	63
11	64,12	0,00	0,00	42	94
14	65,25	0,00	0,00	41	66
6	77,40	0,00	3,40	40	74
15	89,29	0,00	4,29	45	85
3	94,47	0,00	3,47	28	91
13	99,40	0,00	4,40	49	95
1	106,52	0,00	0,00	0	100

FONTE: O autor(2011)

A solução encontrada para  $n = 20$  nós é apresentada na tabela 5.16.

TABELA 5.16: Solução ACS do modelo para 20 nós - CENÁRIO 4

Nó	$s_i$	$D_i^-$	$D_i^+$	$a_i$	$b_i$
1	0,00	0,00	0,00	0	100
6	6,40	2,60	0,00	9	32
15	18,29	0,00	0,00	15	32
20	25,68	0,00	0,00	25	65
11	27,03	0,00	0,00	3	59
19	28,20	0,00	0,00	7	69
3	33,68	0,00	0,00	16	88
17	39,87	0,00	0,00	3	53
10	43,66	0,00	0,00	21	81
14	46,90	0,00	0,00	23	79
7	48,46	0,00	0,46	23	48
13	49,65	0,00	11,65	19	38
4	59,27	0,00	2,27	21	57
12	69,53	0,00	9,53	13	60
5	78,81	0,00	0,00	30	89
18	90,33	0,00	4,33	1	86
8	98,03	0,00	13,03	7	85
16	103,80	0,00	62,80	7	41
2	111,44	0,00	53,44	16	58
9	121,14	0,00	40,14	14	81
1	129,76	0,00	0,00	0	100

FONTE: O autor(2011)

As tabelas 5.17 e 5.18 apresentam a solução para  $n = 40$  nós. Nesta solução foram utilizados 2 veículos, um veículo atende 22 clientes (TABELA 5.17) e outro veículo atende 17 clientes (TABELA 5.18).

TABELA 5.17: Solução ACS do modelo para 40 nós - CENÁRIO 1 - rota 1

Nó	$s_i$	$D_i^-$	$D_i^+$	$a_i$	$b_i$
1	0,00	0,00	0,00	0	100
34	4,37	28,63	0,00	33	62
29	7,00	19,00	0,00	26	45
40	11,13	23,87	0,00	35	69
16	12,35	4,65	0,00	17	41
18	13,26	17,74	0,00	31	68
14	19,18	5,82	0,00	25	62
11	20,30	0,00	0,00	19	69
27	21,44	0,00	0,00	14	41
35	23,31	0,00	0,00	20	61
30	26,43	0,00	0,00	16	60
38	29,11	0,00	0,00	26	41
21	37,39	0,00	0,00	16	43
28	41,31	0,00	1,31	33	40
20	42,04	0,00	1,04	24	41
23	42,78	0,00	0,00	16	59
15	48,26	0,00	0,00	31	58
32	60,42	0,00	0,42	19	60
39	64,63	0,00	18,63	11	46
31	69,64	0,00	1,64	24	68
4	86,61	0,00	50,61	26	36
10	96,52	0,00	47,52	13	49
25	99,48	0,00	30,48	29	69
1	106,41	0,00	0,00	0	100

FONTE: O autor(2011)

TABELA 5.18: Solução ACS do modelo para 40 nós - CENÁRIO 1 - rota 2

Nó	$s_i$	$D_i^-$	$D_i^+$	$a_i$	$b_i$
1	0,00	0,00	0,00	0	100
37	4,92	10,08	0,00	15	63
36	8,03	6,97	0,00	15	69
22	13,99	13,01	0,00	27	37
33	22,53	11,47	0,00	34	56
26	23,91	0,00	0,00	14	38
7	25,15	2,85	0,00	28	57
9	26,51	0,00	0,00	13	46
24	32,54	0,00	0,00	18	46
2	41,11	0,00	0,00	27	68
8	48,56	0,00	0,00	18	65
3	53,12	0,00	0,00	19	56
19	58,60	0,00	13,60	18	45
13	68,52	0,00	32,52	12	36
5	77,23	0,00	10,23	35	67
12	86,51	0,00	41,51	31	45
6	97,50	0,00	46,50	28	51
17	111,73	0,00	42,73	27	69
1	120,25	0,00	0,00	0	100

FONTE: O autor(2011)

Um resumo das soluções encontradas para as instâncias testadas para a meta-heurística ACS é mostrado na tabela 5.19. Observa-se que em algumas situações, embora o número de nós seja menor, a solução encontrada apresentou um custo superior às instâncias maiores. Isso deve-se ao fato do custo total variar em função da distância percorrida pelos veículos e dos desvios de janela. Para  $n = 35$ , nós 71% dos clientes foram atendidos fora de sua janela de tempo, enquanto que para  $n = 45$  apenas 47% dos clientes foram atendidos com desvio.

Observa-se também que o tempo computacional gasto foi satisfatório visto que para  $n = 50$  nós o tempo de execução foi aproximadamente 74 vezes menor que o tempo de execução da solução exata para  $n = 20$  nós.

TABELA 5.19: Resumo das soluções obtidas com a meta-heurística ACS

n	Cenário	Custo total	Veículos utilizados	Clientes atendidos fora da janela	Tempo de execução(s)
10	1	260,32	1	6	11
	2	220,14	1	3	10
	3	243,19	1	7	15
	4	206,41	1	6	7
15	1	226,97	1	7	195
	2	215,20	1	6	205
	3	237,12	1	5	222
	4	218,21	1	5	210
20	4	297,90	1	10	389
25	1	289,14	1	11	402
30	2	681,93	2	19	453
35	3	1823,38	2	25	498
40	4	886,82	2	25	553
45	1	766,46	2	21	589
50	2	1173,54	2	30	611

FONTE: O autor(2011)

### 5.3 COMPARAÇÃO DOS RESULTADOS EXATOS E HEURÍSTICOS

Nesta seção são comparados as soluções do modelo PRVER encontradas utilizando-se a formulação exata do problema e através da meta-heurística *Ant Colony System*.

Na tabela 5.20 estão apresentadas as soluções exata e ACS para  $n = 10$  nós. Nota-se que a solução aproximada via ACS foi igual a solução exata.

TABELA 5.20: Soluções exata e ACS para 10 nós - CENÁRIO 3

SOLUÇÃO EXATA				SOLUÇÃO ACS			
Nó	$s_i$	$D_i^-$	$D_i^+$	Nó	$s_i$	$D_i^-$	$D_i^+$
1	0,00	0,00	0,00	1	0,00	0,00	0,00
9	8,62	12,38	0,00	9	8,62	12,38	0,00
4	20,96	7,04	0,00	4	20,96	7,04	0,00
7	33,27	0,00	0,00	7	33,27	0,00	0,00
2	43,05	6,95	0,00	2	43,05	6,95	0,00
3	49,78	0,22	0,00	3	49,78	0,22	0,00
8	54,34	0,00	0,00	8	54,34	0,00	0,00
5	62,40	0,00	5,40	5	62,40	0,00	5,40
10	71,39	0,00	19,39	10	71,39	0,00	19,39
6	82,07	0,00	17,07	6	82,07	0,00	17,07
1	89,01	0,00	0,00	1	89,01	0,00	0,00

FONTE: O autor(2011)

A tabela 5.21 apresenta a comparação entre as soluções exata e ACS para  $n = 15$ . A solução ACS foi diferente da solução exata em dois pontos da rota, como indicado nas linhas em destaque.

TABELA 5.21: Soluções exata e ACS para 15 nós - CENÁRIO 4

SOLUÇÃO EXATA				SOLUÇÃO ACS			
Nó	$s_i$	$D_i^-$	$D_i^+$	Nó	$s_i$	$D_i^-$	$D_i^+$
1	0,00	0,00	0,00	1	0,00	0,00	0,00
6	6,40	21,60	0,00	6	6,40	21,60	0,00
13	16,78	0,00	0,00	13	16,78	0,00	0,00
4	26,40	0,00	0,00	4	26,40	0,00	0,00
15	36,66	0,00	0,00	14	36,99	0,00	0,00
11	37,76	0,00	0,00	11	38,12	0,00	0,00
14	38,88	0,00	0,00	15	39,22	0,00	0,00
7	40,44	0,00	0,00	7	40,74	0,00	0,00
9	41,80	0,00	0,00	9	42,10	0,00	0,00
3	47,74	0,00	0,00	3	48,04	0,00	0,00
10	52,71	0,00	3,71	10	53,01	0,00	4,01
2	60,84	0,00	0,00	2	61,14	0,00	0,00
8	68,30	0,00	3,30	8	68,60	0,00	3,60
5	76,36	0,00	9,36	5	76,66	0,00	9,66
12	85,63	0,00	40,63	12	85,93	0,00	40,93
1	93,08	0,00	0,00	1	93,38	0,00	0,00

FONTE: O autor(2011)

Para  $n = 20$  nós observa-se que a solução ACS diferiu da solução exata em 4 pontos na rota (TABELA 5.22).

TABELA 5.22: Soluções exata e ACS para 20 nós - CENÁRIO 4

SOLUÇÃO EXATA				SOLUÇÃO ACS			
Nó	$s_i$	$D_i^-$	$D_i^+$	Nó	$s_i$	$D_i^-$	$D_i^+$
1	0,00	0,00	0,00	1	0,00	0,00	0,00
6	6,40	2,60	0,00	6	6,40	2,60	0,00
15	18,29	0,00	0,00	15	18,29	0,00	0,00
20	25,68	0,00	0,00	19	24,86	0,00	0,00
11	27,03	0,00	0,00	10	28,22	0,00	0,00
19	28,20	0,00	0,00	20	32,09	0,00	0,00
3	33,68	0,00	0,00	3	37,99	0,00	0,00
17	39,87	0,00	0,00	17	44,18	0,00	0,00
10	43,66	0,00	0,00	11	45,49	0,00	0,00
14	46,90	0,00	0,00	14	46,61	0,00	0,00
7	48,46	0,00	0,46	7	48,17	0,00	0,17
13	49,65	0,00	11,65	13	49,37	0,00	11,37
4	59,27	0,00	2,27	4	58,99	0,00	1,99
12	69,53	0,00	9,53	12	69,25	0,00	9,25
5	78,81	0,00	0,00	5	78,53	0,00	0,00
18	90,33	0,00	4,33	18	90,05	0,00	4,05
8	98,03	0,00	13,03	8	97,75	0,00	12,75
16	103,80	0,00	62,80	16	103,52	0,00	62,52
2	111,44	0,00	53,44	2	111,16	0,00	53,16
9	121,14	0,00	40,14	9	120,86	0,00	39,86
1	129,76	0,00	0,00	1	129,48	0,00	0,00

FONTE: O autor(2011)

Na tabela 5.23 é apresentada uma comparação entre o custo total da solução exata e da solução ACS, e o erro relativo relacionado a aproximação.

$$\text{Erro relativo} = \frac{|Sol_{EXATA} - Sol_{ACS}|}{Sol_{EXATA}}$$

Observa-se que para todas as instâncias de  $n = 10$  e para uma instância de  $n = 15$  a solução ACS foi igual à solução exata para todas as instâncias testadas. Nas demais instâncias testadas a solução ACS apresentou uma pequena diferença em relação à solução exata.

TABELA 5.23: Comparação entre as soluções exata e ACS

n	Cenário	Solução exata	Solução ACS	Erro relativo percentual
10	1	260,32	260,32	0,0 %
	2	220,14	220,14	0,0 %
	3	243,19	243,19	0,0 %
	4	206,41	206,41	0,0 %
15	1	225,92	226,97	0,5 %
	2	213,84	215,20	0,6 %
	3	237,12	237,12	0,0 %
	4	215,72	218,21	1,2 %
20	4	294,93	297,90	1,0 %

FONTE: O autor(2011)

Uma comparação entre os tempos de execução da solução exata e da solução ACS é apresentada na tabela 5.24. Conforme esperado o tempo computacional gasto pela meta-heurística ACS é significativamente menor que o tempo gasto para encontrar a solução exata.

TABELA 5.24: Comparação entre os tempos de execução para as soluções exata e ACS

n	Cenário	Tempo Solução exata	Tempo Solução ACS	Diferença (%)
10	1	32	11	65,62
	2	30	10	66,67
	3	28	15	46,43
	4	31	7	77,42
15	1	14450	195	98,65
	2	14423	205	98,58
	3	18010	222	98,76
	4	18025	210	98,83
20	4	45652	389	99,14

FONTE: O autor(2011)

Os resultados apresentados nas seções 5.2.1 e 5.2.2 mostram que o modelo proposto se adequou ao caso estudado: os desvios de janela foram minimizados bem como a distância percorrida pelos veículos. Além disso, conforme apresentado na seção 5.3 a meta-heurística *Ant Colony System* implementada para a resolução do modelo mostrou-se eficiente, visto que apresentou soluções de boa qualidade em tempo computacional viável.

#### 5.4 ANÁLISE COMPARATIVA ENTRE OS CASOS DETERMINÍSTICO E ESTOCÁSTICO

Nesta seção é feita uma comparação entre os resultados encontrados para os casos determinístico e estocástico do problema de entrega de refeições, a fim de analisar o quanto a variabilidade da demanda e dos clientes afeta os roteiros de entrega.

No capítulo 4 foram apresentados os resultados para o problema determinístico, ou seja, as rotas foram geradas a partir do conhecimento prévio de todos os dados. No capítulo 5 foram apresentados os resultados para o problema estocástico: as rotas foram geradas a partir da resolução do modelo PRVER, os parâmetros deste modelo são baseados no histórico de pedidos dos clientes durante o ano de 2009.

Na tabela 5.25 são comparadas as rotas de entrega para os casos estocástico e determinístico considerando-se  $n = 10$  nós. Os roteiros foram iguais somente em dois pontos, como indicam as linhas em destaque na tabela.

TABELA 5.25: Comparação entre os roteiros estocástico e determinístico - 10 nós

ROTEIRO ENCONTRADO	
Caso estocástico	Caso determinístico
1	1
6	6
10	4
5	5
8	7
4	10
7	9
3	8
2	3
9	2
1	1

FONTE: O autor(2011)

Considerando-se as rotas encontradas nas seções 5.2.1 e 5.2.2 foi calculada a distância real percorrida pelos veículos no caso estocástico. Isto foi feito pois o valor total da solução apresentado nestas seções é um valor ponderado que considera além da distância percorrida, a probabilidade de presença dos clientes e os desvios de janela. A tabela 5.26 apresenta uma comparação entre a distância real percorrida

pelos veículos nos casos estocástico e determinístico.

TABELA 5.26: Comparação entre os resultados estocásticos e determinísticos

n	Cenário	Distância real percorrida (km)		Diferença (%)
		Caso estocástico	Caso determinístico	
10	1	38,860	37,155	4,589
15	2	51,975	45,785	13,520
20	3	63,520	53,755	18,166
25	1	68,740	59,495	15,539
30	2	93,225	74,515	25,109
35	3	90,140	82,525	9,228
40	4	95,425	86,215	10,683
45	1	100,805	91,955	9,624
50	2	107,445	95,405	12,620

FONTE: O autor(2011)

A figura 5.3 apresenta a distância total percorrida nas rotas obtidas através dos métodos estocástico e determinístico. A distância total encontrada pela solução determinística foi em média 13,30% menor que a distância total encontrada pela solução estocástica.

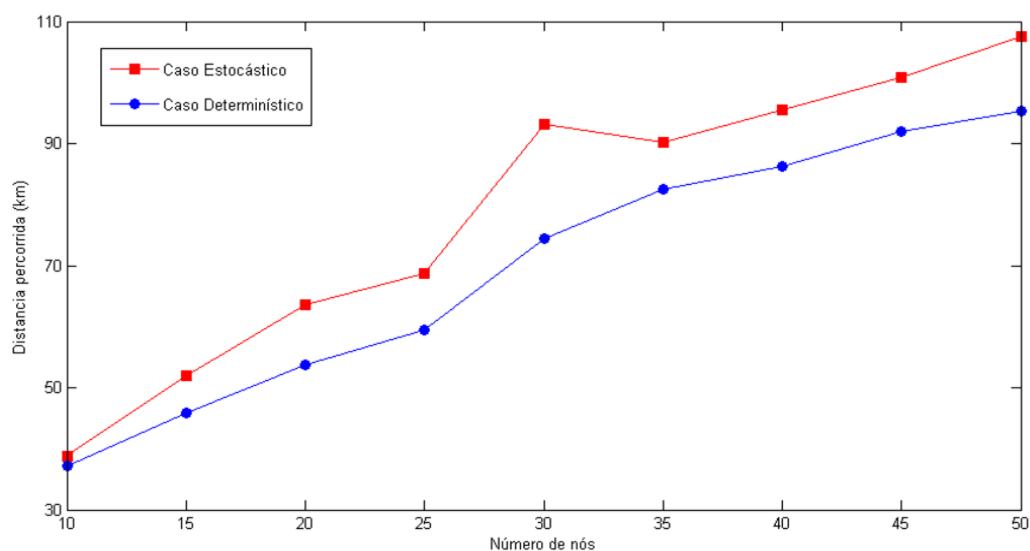


FIGURA 5.3: Distância percorrida: solução estocástica e solução determinística

FONTE: O autor(2011)

Analisando-se a tabela 5.26 percebe-se que há uma diferença significativa entre as soluções estocástica e determinística para  $n = 20, 25, 30$  nós o que indica a necessidade de ajuste do modelo PRVER para que este apresente melhores soluções.



## 6 CONSIDERAÇÕES FINAIS E SUGESTÕES PARA TRABALHOS FUTUROS

Neste trabalho foram apresentadas uma abordagem determinística e uma abordagem estocástica para propor uma solução ao problema de roteamento de veículos com janela de tempo e demanda estocástica, no caso específico de entrega de refeições feitas pela ASSEMCO - Associação dos Servidores Municipais de Colombo.

Na abordagem determinística foi desenvolvido o programa computacional GERAROTA para criação dos roteiros diários de entrega. Neste programa foram implementadas as heurísticas de Teitz e Bart (1968) e Gillet e Johnson (1974) para formação dos grupos de atendimento, e as heurísticas do vizinho mais próximo e 2-opt para construção das rotas em cada grupo de atendimento.

Os resultados obtidos com o uso do programa GERAROTA se mostraram satisfatórios em relação aos roteiros normalmente praticados pelos funcionários. Ao minimizar a distância percorrida pelos veículos, consequentemente obteve-se uma diminuição no tempo de realização das entregas e de permanência da refeição dentro do veículo, o que sem dúvida é fundamental para conservação da qualidade da mesma.

Na abordagem estocástica foi proposto um modelo que considera a probabilidade de presença dos clientes, que neste estudo de caso significa que o cliente fez um pedido, a probabilidade de utilização do arco  $(i, j)$ , calculada considerando-se a probabilidade de presença dos nós  $i$  e  $j$ , e dos nós intermediários, e ainda, o tempo de viagem entre os clientes, baseado na probabilidade de presença destes. Além disso, o modelo permite que o cliente seja atendido fora da janela de tempo determinada, neste caso é considerado um desvio da janela (por atraso ou antecedência) e uma penalidade é imposta. O objetivo do modelo é minimizar os desvios da janela, a distância percorrida pelos veículos e o número de veículos utilizados.

Observou-se que o modelo foi eficiente pois os desvios de janela foram minimizados bem como a distância percorrida pelos veículos e o número de veículos utilizados. Em virtude da complexidade computacional do modelo este pôde ser resolvido de forma exata apenas para instâncias pequenas, e por essa razão foi implementada a meta-heurística *Ant Colony System* (ACS). Esta apresentou soluções de boa qualidade em tempo computacional significativamente menor que o tempo gasto para encontrar a solução exata, mesmo nos casos onde não seria possível encontrar a solução exata.

Como sugestões para trabalhos futuros propõe-se aplicar esta metodologia em outras situações de atendimento à clientes. Utilizar heurísticas de melhoramento de rotas junto com a meta-heurística ACS, ou ainda, gerar uma solução inicial para o ACS a fim de buscar soluções melhores em menor tempo computacional.

## REFERÊNCIAS

- APPLEGATE, D. *et al.* **Optimal tour of sweden**. 2004. [Online; acesso em 17-Outubro-2010]. Disponível em: <<http://www.tsp.gatech.edu/sweden/>>.
- ATKINSON, J. A vehicle-scheduling system for delivering school meals. **Journal of Operational Research Society**, v. 41, n. 8, p. 703–711, 1990.
- AZI, N.; GENDREAU, M.; POTVIN, J.-Y. An exact algorithm for a single vehicle routing problem with time windows and multiple route. **European Journal of Operational Research**, v. 178, p. 755–766, 2007.
- BAKER, M.; AYECHIEW, M. A genetic algorithm for the vehicle routing problem. **Computers and Operations Research**, v. 20, p. 787–800, 2003.
- BALLOU, R. H. **Gerenciamento da Cadeia de Suprimentos: planejamento, organização e logística empresarial**. 4<sup>a</sup>. ed. Porto Alegre: Bookman, 2001.
- BARBAROSOGLU, G.; OZGUR, D. A tabu search algorithm for the vehicle routing problem. **Computers and Operations Research**, v. 26, p. 255–270, 1999.
- BARD, J. F. Validating vehicle routing zone construction using monte carlo simulation. **European Journal of Operational Research**, 2010.
- BARTHOLDI, J. *et al.* A minimal technology routing system for meals and wheels. **Interfaces**, v. 13, p. 1–8, 1983.
- BELTRAMI, E.; BODIN, L. Networks and vehicle routing for municipal waste collection. **Networks**, v. 4, p. 65–94, 1974.
- BERGER, J.; BARKAOUI, M.; BRÄSSY, O. **A parallel hybrid genetic algorithm for the vehicle routing problem with time windows**. Canada, 2001.
- BODIN, L. *et al.* Routing and scheduling of vehicles and crews: the state of the art. **Computers and Operations Research**, v. 10, p. 69–211, 1983.
- BRAMEL, J.; SIMCHI-LEVI, D. Probabilistic analyses and practical algorithms for the vehicle routing problem with time windows. **Operational Research**, v. 44, p. 501–509, 1993.
- BRÄSSY, O. A reactive variable neighborhood search for the vehicle-routing problem with time windows. **Inform Journal on computing**, v. 15(4), p. 347–368, 2003.
- BRÄSSY, O.; GENDREAU, M. A tabu search heuristic for the vehicle routing problem with time windows. **Sociedad de Estadística e Investigación Operativa**, v. 10(2), p. 211–237, 2002.

BRÄYSY, O. *et al.* An optimization approach for communal home meal delivery service: A case study. **Journal of Computational and Applied Mathematics**, v. 232, p. 46–53, 2008.

BULLNHEIMER, B.; HARTL, R.; STRAUSS, C. An improved ant system algorithm for the vehicle routing problem. **Annals of Operations Research**, v. 89, p. 319–328, 1999.

CHIANG, W.; RUSSEL, R. A. Simulated annealing metaheuristics for the vehicle routing problem with time windows. **Annals of Operations Research**, v. 63, p. 3–27, 1996.

CHRISTOFIDES, N.; MINGOZZAIN, A.; TOTH, P. Exact algorithms for the vehicle routing problem based on the spanning tree and shortest path relaxations. **Math. Programming**, v. 20, p. 255–282, 1981b.

CHRISTOFIDES, N.; MINGOZZI, A.; TOTH, P. Exact algorithms for the vehicle routing problem based on spanning tree and shortest path relaxation. **Math. Programming**, v. 10, p. 255–280, 1981a.

COOK, W.; RICH, J. **A Parallel Cutting-Plane Algorithm for the Vehicle Routing problems with Time Windows**. Houston, 1999.

COVER, T.; HART, P. E. Nearest neighbor pattern classification. **IEEE Transaction on Information Theory**, v. 13(1), p. 21–27, 1967.

CROWDER, H.; PADBERG, M. W. Solving large-scale symmetric traveling salesman problems to optimality. **Management Science**, v. 26, p. 495–509, 1980.

CZECH, Z. J.; CZARNAS, P. A parallel simulated annealing for the vehicle routing problem with time windows. **Proceedings 10th Euromicro Workshop on Parallel, Distributed and Network-based Processing**, p. 376–383, 2002.

DANTZIG, G. B.; FULKERSON, R.; JOHNSON, S. M. Solution of a large-scale traveling salesman problem. **Operations Research**, v. 2, p. 393–410, 1954.

DANTZIG, G. B.; RAMSER, R. The truck dispatching problem. **Management Science**, v. 6, p. 80–91, 1959.

DENEUBOURG, J.-L. *et al.* The self-organizing exploratory pattern of the argentine ant. **Journal of Insect Behavior**, v. 3, p. 159–168, 1990.

DESROCHERS, M. Vehicle routing with time windows: Optimization and approximation. In: **Vehicle Routing: Methods and Studies**. Amsterdam: Elsevier Science Publishers, 1988. p. 65–84.

DESROCHERS, M.; DESROSIERS, J.; SOLOMON, M. A new optimization algorithm for the vehicle routing problem with time windows. **Operations Research**, v. 40, p. 342–354, 1992.

DORIGO, M.; CARO, G. D.; GAMBARDELLA, L. M. Ant algorithms for discrete optimization. **Artificial Life**, v. 5(2), p. 137–172, 1999.

DORIGO, M.; MANIEZZO, V.; COLORNI, A. Ant system: Optimization by a colony of cooperating agents. **IEEE Transactions on Systems, Man, and Cybernetics Part B**, v. 26(1), p. 29–41, 1996.

DORIGO, M.; STÜTZLE, T. **Ant Colony Optimization**. Massachusetts: MIT Press, 2004.

FEILLET, D. e. a. An exact algorithm for the elementary shortest path problem with resource constraints: Application to some vehicle routing problems. **Networks**, v. 44, p. 216–229, 2004.

FISHER, M.; JAIKUMAR, R. A generalized assignment heuristics for vehicle routing. **Networks**, v. 11, p. 109–124, 1981.

GAMBARDELLA, L.; TAILLARD, E.; AGAZZI, G. Macs-vrptw: A multiple ant colony system for vehicle routing problems with time windows. In: **New Ideas in Optimization**. [S.l.]: McGraw-Hill, 1999. p. 63–76.

GAMBARDELLA, L. M.; DORIGO, M. Solving symmetric and asymmetric tsps by ant colonies. **Proceedings of the 1996 IEEE International Conference on Evolutionary Computation**, p. 622–627, 1996.

GENDRAU, M.; LAPORTE, G.; SÉGUIN, R. Stochastic vehicle routing. **European Journal of Operational Research**, p. 3–12, 1996.

GENDREAU, M. Recent advances in stochastic vehicle routing - conference. **XLII Simpósio Brasileiro de Pesquisa Operacional**, 2010.

GILLET, B.; JOHNSON, J. Sweep algorithm for the multiple depot vehicle dispatch problem. **Proceedings of ORSA/TIMS Meeting**, 1974.

GILLET, B.; MILLER, L. A heuristic algorithm for the vehicle dispatch problem. **Operations Research**, v. 22, p. 340–349, 1974.

GOLDBARG, M.; LUNA, H. **Otimização combinatória e programação linear: modelos e algoritmos**. 2<sup>a</sup>. ed. Rio de Janeiro: Elsevier, 2005.

GOLDEN, B.; MAGNANTI, T.; NGUYEN, H. Implementing vehicle routing algorithms. **Networks**, v. 7, p. 113–148, 1977.

HALL, R.; PARTYKA, J. On the road to efficiency. **OR/MS Today**, p. 38–47, 1997.

HAUGLAND, D.; HO, S.; LAPORTE, G. Designing delivery districts for the vehicle routing problem with stochastic demands. **European Journal of Operational Research**, v. 180, p. 997–1010, 2007.

HOMBERGER, J.; GEHRING, H. Two evolutionary metaheuristic for the vehicle routing problems with time windows. **Inform Systems Operational Research**, v. 37, p. 297–318, 1999.

IOANNOU, G. A greedy look-ahead heuristic for the vehicle routing with time windows. **Journal of Operations Research Society**, v. 52, p. 523–537, 2001.

JOHNSON, M.; ROEHRIG, S.; YILDIZ, H. **A Genetic Algorithm for the Home-Delivered Meals Location-Routing Problem**. Pittsburgh, 2009.

KOHL, N.; MADSEN, B. An optimization algorithm for the vehicle routing problem with time windows based on lagrangean relaxation. **Operations Research**, v. 45, p. 395–406, 1997.

KOLEN, A. J.; KAN, A. H. G. R.; TRIENEKENS, H. W. J. Vehicle routing with time windows. **Operations Research**, v. 35, p. 266–273, 1987.

KONTORAVDIS, G.; BARD, J. A grasp for the vehicle routing problem with time windows. **Inform Journal on computing**, v. 7(1), p. 10–23, 1995.

LAPORTE, G.; NOBERT, Y.; DESROCHERS, M. Optimal routing under capacity and distance restrictions. **Operations Research**, v. 33, p. 1050–1073, 1985.

LARSEN, A. **The dynamic vehicle routing**. Tese (Doutorado) — Department of Mathematical Modeling, Technical University of Denmark, 2000.

LEE, L. *et al.* Vehicle capacity planning system: A case study on vehicle routing problem with time windows. **IEEE Transactions on Systems, Man and Cybernetics, Part A**, v. 33(2), 2003.

LI, H.; LIM, A. Local search with annealing-like restarts to solve the vrptw. **European Journal of Operational Research**, v. 150, p. 115–127, 2003.

LIN, S. Computer solutions of the traveling salesman problem. **The Bell System Technical Journal**, v. 44, 1965.

LORENZI, D. **Uma abordagem híbrida baseada em regras de despacho e busca tabu para o problema de roteamento de veículos com coletas e entregas**. Dissertação (Mestrado) — Pontifícia Universidade Católica do Paraná, 2003.

MARK, K.; GUO, Z. A genetic algorithm for vehicle routing problems with stochastic demand and soft time windows. **Proceedings of the 2004 Systems and Information Engineering Design Symposium**, p. 183–190, 2004.

NEWTON, R.; THOMAS, W. Bus routing in multischool system. **Computers and Operational Research**, v. 1, p. 213–222, 1974.

PADBERG, M.; RINALDI, G. A branch-and-cut algorithm for the resolution of large-scale symmetric travelling salesman problems. **SIAM Review**, v. 33, p. 60–100, 1991.

PAVONE, M. *et al.* A stochastic and dynamic vehicle routing problem with time windows and customer impatience. **ACM/Springer Mobile Networks and Applications Journal**, v. 14, n. 3, p. 350–364, 2009.

PELLEGRINI, P. **Application of two nearest neighbor approaches to a rich vehicle routing problem**. Bruxelas, 2005.

POTVIN, J.; ROUSSEAU, J. M. A parallel route building algorithm for the vehicle routing and scheduling problem with time windows. **European J. Operations Research**, v. 66, p. 331–340, 1993.

PRINS, C. A simple and effective evolutionary algorithm for the vehicle routing problem. **Computers and Operations Research**, v. 31, p. 1985–2002, 2004.

RALPHS, T. K. *et al.* On the capacitated vehicle routing problem. **Math. Programming**, v. 94, p. 343–359, 2003.

REI, W.; GENDREAU, M.; SORIANO, P. A hybrid monte carlo local branching algorithm for the single vehicle routing problem with stochastic demands. **Transportation Science**, v. 44, n. 1, p. 136–146, 2010.

REIMANN, M.; STUMMER, M.; DOERNER, K. A savings based ant system for the vehicle routing problem. **GECCO 2002: Proceedings of the Genetic and Evolutionary Computation Conference**, 2002.

REINELT, G. **TSPLIB: biblioteca de instâncias para o PCV**. 1990. [Online; acesso em 10-Outubro-2010]. Disponível em: <<http://www2.iwr.uni-heidelberg.de/groups/comopt/software/TSPLIB95/>>.

ROUSSEAU, L.; GENDREAU, M.; PESANT, G. Solving vrptw's with constraint programming based column generation. **Annals of Operations Research**, v. 130, p. 199–216, 2004.

RUSSEL, R. Hybrid heuristics for the vehicle routing problem with time windows. **Transportation Science**, v. 29, p. 156–166, 1995.

SECOMANDI, N.; MARGOT, F. Reoptimization approaches for the vehicle-routing problem with stochastic demands. **Operations research**, v. 57, n. 1, p. 214–230, 2009.

SOLOMON, M. Algorithms for vehicle routing and scheduling problems with time window constraints. **Operations Research**, v. 35(2), p. 254–266, 1987.

SOLOMON, M.; DESROCHERS, J. Time windows constrained routing and scheduling problems. **Transportation Science**, v. 22, p. 1–13, 1988.

SUENSILPONG, T. **Vehicle routing system for daily meal delivery with motorcycles**. Dissertação (Mestrado) — Chulalongkorn University, Thailand, 2007.

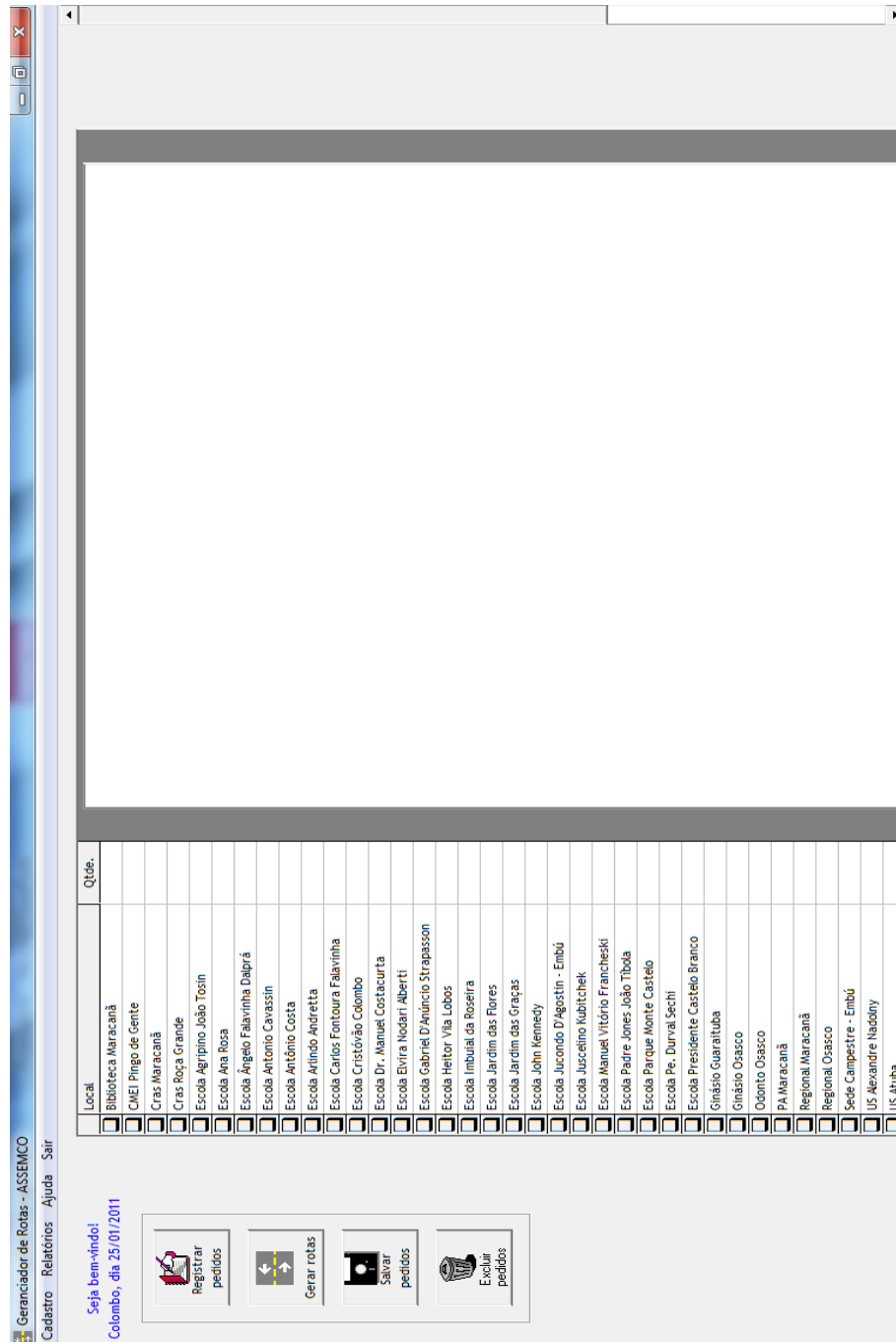
TAN, K.; CHEONG, C.; GOH, C. Solving multiobjective vehicle routing problem with stochastic demand via evolutionary computation. **European Journal of Operational Research**, v. 177, p. 813–839, 2007.

TEITZ, M.; BART, P. Heuristics methods for estimating the generalized vertex median of a weighted graph. **Operations Research**, 1968.

TIMLIN, M.; PULLEYBLANK, W. **Procedence constrained routing and helicopter scheduling: heuristic design**. Canada, 1990.

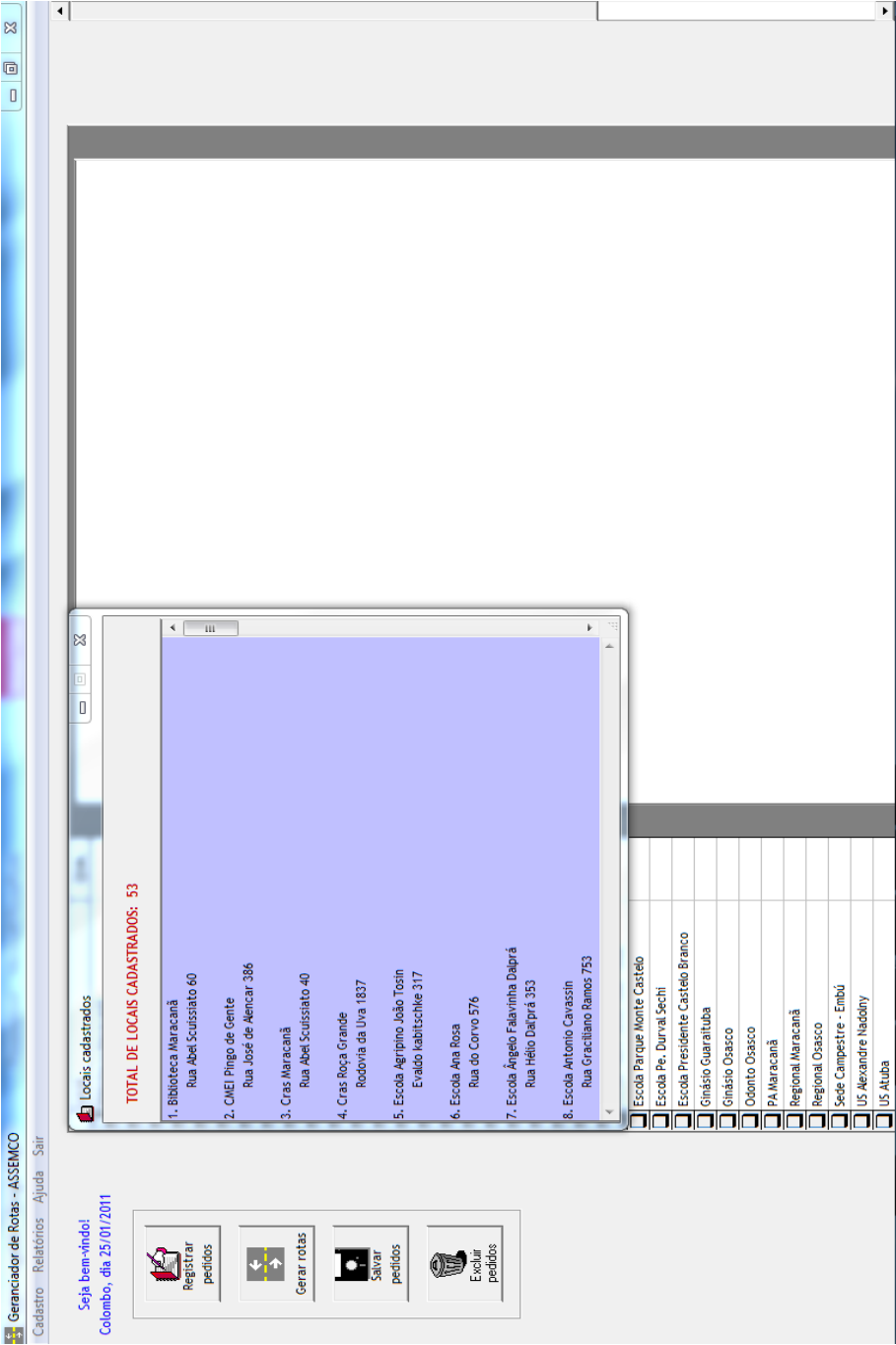
TING, C. J.; HUANG, C. H. An improved genetic algorithm for vehicle routing problem with time windows. **International Journal of Industrial Engineering**, 2004.

## APÊNDICE A – JANELA PRINCIPAL - PROGRAMA GERAROTA





APÊNDICE B – JANELA DE LOCAIS CADASTRADOS - PROGRAMA GERAROTA



## APÊNDICE C – JANELA COM O TRAJETO SUGERIDO - PROGRAMA GERAROTA

Roteiro sugerido - Google Maps

É necessário estar conectado à Internet para visualizar o trajeto de ruas sugerido.

Web | Imagens | Vídeos | Mapas | Notícias | Livros | Gmail | mais ▼

Novidade! | Ajuda | Fazer login

Google maps

José Leal Fontoura 113 Colombo Paraná 83414-190

Pesquisar no Mapa

Mostrar opções de pesquisa

Como chegar

Meus mapas

Imprimir | Enviar | Link

Satélite

Trânsito

Rota de carro para R. do Pinheiro, 140 - Colombo - PR, 83414-570, Brasil

Adicionar destino - Mostrar opções

Como chegar

A José Leal Fontoura 113 Colombo Paraná 83414-190  
 B Rua Prefeito João Batista Stocco Colombo Paraná 83414-190  
 C Rua Angelina Cavalli 292 Colombo Paraná 83408-100  
 D Rua Graciliano Ramos 753 Colombo Paraná 83414-190  
 E Rua Abel Scussiato 60 Colombo Paraná 83408-100  
 F Travessa Roma 35 Colombo Paraná 83413-280  
 G Travessa São Paulo 141 Colombo Paraná 83404-100  
 H Rua Santo Paschoal Franceschi Colombo Paraná 83404-100  
 I Rodovia da Uva 1837 Colombo Paraná 83404-100  
 J Rua Milton Tosin 64 Colombo Paraná 83402-420  
 K Rua Colibri 31 Colombo Paraná 83401-180  
 L Rua do Pinheiro 140 Colombo Paraná 83414-570

## ANEXO A – DISTRIBUIÇÕES DE PROBABILIDADE

### Distribuição de Poisson

Uma variável aleatória discreta  $X$  tem uma Distribuição de Poisson quando sua função de probabilidade é dada por:

$$P(X = x) = \frac{\theta^x \cdot e^{-\theta}}{x!} \quad x = 0, 1, 2, \dots \quad e \quad \theta > 0$$

A esperança e a variância de  $X$  são  $E(X) = \mu = \theta$  e  $V(X) = \sigma^2 = \theta$

### Distribuição Normal

Uma variável aleatória contínua  $X$  é normalmente distribuída quando sua função densidade de probabilidade é da forma:

$$f(x) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{1}{2}\left(\frac{x-\mu}{\sigma}\right)^2} \quad x \in \Re, \mu \in \Re, \sigma \in \Re$$

A esperança e a variância da variável aleatória  $X$  são  $E(X) = \mu$  e  $V(X) = \sigma^2$ , respectivamente.

### Distribuição Gama

Uma variável aleatória contínua  $X$  tem uma Distribuição Gama com parâmetros  $\alpha$  e  $\beta$  quando sua função densidade de probabilidade é dada por:

$$f(x) = \begin{cases} \frac{\beta^\alpha}{\Gamma(\alpha)} x^{\alpha-1} e^{-\beta x}, & x > 0 \\ 0, & x \leq 0 \end{cases}$$

onde a função gama é definida por  $\Gamma(n) = \int_0^\infty t^{n-1} e^{-t} dt, n > 0$ .

A esperança e a variância de  $X$  são dadas por  $E(X) = \frac{\alpha}{\beta}$  e  $V(X) = \frac{\alpha}{\beta^2}$ .

### **Distribuição Uniforme**

Uma variável aleatória contínua  $X$  é uniformemente distribuída em um intervalo  $[a, b]$  se sua função densidade de probabilidade é dada por:

$$f(x) = \frac{1}{b-a}$$

A esperança e a variância de  $X$  são dadas por  $E(X) = \mu = \frac{a+b}{2}$  e  $V(X) = \sigma^2 = \frac{(b-a)^2}{12}$ .